NEXTFLICK – AN LLM-BASED MOVIE RECOMMENDATION SYSTEM

Harsh A. Dudhe[1], Parth A. Ingole[2], Vedika V. Ade[3], Saloni J. Kuldipkar[4],Sachin D. Chavhan[5]
[1,2,3,4] *Student,* [5]*Faculty of Department of Computer Science and Engineering*
Babasaheb Naik College of Engineering (BNCOE), Pusad-445204
Maharashtra , India

Abstract:
With the rapid growth of digital entertainment, viewers today are overwhelmed by the vast number of movies and shows available online. Choosing what to watch often becomes time-consuming, as users scroll through endless options without finding something that matches their mood or preference. Current recommendation systems mainly rely on popularity trends, content-based filtering, or collaborative filtering. While useful, these approaches are limited. They often provide generic suggestions, struggle with new users (the cold start problem), and fail to capture complex or diverse tastes. This makes the recommendation process less satisfying and highlights the need for smarter, more personalized solutions. The purpose of this paper is to develop Nextflick, a movie recommendation system that goes beyond traditional methods. Our approach will integrate natural language understanding with existing recommendation algorithms. Instead of only showing predefined lists, the system will allow users to express their needs in everyday language. The system will process these queries, identify the intent, and provide ranked recommendations that feel more relevant and user-friendly. Our planned process is to use, Natural Language Processing (NLP) through Large Language Models (LLMs). To generate recommendations, we will apply a mix of content-based filtering, collaborative filtering, and hybrid techniques. Through this paper, our aim is to create a system that not only helps users discover movies quickly but also demonstrates how AI and machine learning can improve personalization. Nextflick will serve as both a practical solution and a valuable learning experience for our team.
Keywords: *Natural Language Processing (NLP), Large Language Models (LLMs), Query, Recommendation system.*

1. Introduction

Movies have always been an essential part of human entertainment, education, and culture, serving not only as a source of leisure but also as a medium for storytelling, inspiration, and social reflection. Over the years, cinema has transformed from traditional theatres to home entertainment and now to large-scale digital streaming platforms, allowing users to access thousands of titles at their fingertips. With this exponential growth of digital platforms such as Netflix, Amazon Prime, and Disney+, the number of movies available to audiences has expanded dramatically. While this abundance of choices has enriched the viewing experience, it has also created a new challenge—content overload. Users are often overwhelmed by the sheer volume of options, making it increasingly difficult to find movies that align with their unique tastes, moods, or situational preferences. Recommendation systems have evolved as a solution to this issue. Traditional recommendation engines typically rely on collaborative filtering, popularity-based suggestions, or simple content matching. While these systems can provide a starting point, they suffer from certain limitations. For instance, recommendations may be biased toward trending or widely watched movies, ignoring niche preferences. Similarly, users with limited watch histories may receive generic suggestions that fail to capture their individuality. To bridge this gap, NextFlick is designed as an advanced movie recommendation system that leverages the latest advancements in Large Language Models (LLMs) and Natural Language Processing (NLP). Unlike conventional approaches, NextFlick allows users to interact naturally with the system by entering free-form queries such as *"Suggest me a thrilling detective movie with a strong female lead"* or *"I want to watch something light-hearted and funny like The Intern."* The system interprets these queries, extracts key intent and contextual preferences, and maps them to precise movie recommendations.

This conversational and semantic understanding of user intent significantly enhances personalization, making the recommendation process intuitive and highly accurate.

In essence, the development of NextFlick emphasizes the growing importance of AI-driven personalization in an era where digital content is abundant. By combining the power of LLMs with intelligent recommendation techniques, the platform aims to reduce decision fatigue, enhance user satisfaction, and transform the way audiences discover movies. Ultimately, NextFlick aspires to move beyond traditional recommendation models and represent the next generation of intelligent, human-centric entertainment platforms.

Netflix	2/3rd of the movies watched are recommended
Google News	recommendations generate 38% more click-troughs
Amazon	35% sales from recommendations
Choicestream	28% of the people would buy more music if they found what they liked

Table1. Companies benefit through recommendation system

2. Literature Review

Over the past two decades, movie recommendation systems have undergone significant evolution, shaped by advancements in machine learning, data mining, and artificial intelligence. Several approaches have been proposed, each with distinct methodologies, strengths, and challenges.

1. Collaborative Filtering (CF) Breese et al. (1998) were among the pioneers to propose collaborative filtering for recommendation tasks. The underlying idea is that users with similar preferences in the past are likely to have similar preferences in the future. CF can be categorized into user-based and item-based approaches. While this technique provides effective recommendations when sufficient data is available, it suffers from cold-start problems (new users or new movies with insufficient ratings) and sparsity issues (incomplete rating matrices). Despite these drawbacks, CF laid the foundation for modern recommender systems.

2. Content-Based Filtering (CBF) Panniello et al. (2014) highlighted content-based approaches where recommendations are generated by analysing the attributes of movies (e.g., genre, director, cast, keywords, plot summaries). This method personalizes recommendations for individual users without requiring data from others, thus avoiding some of the cold-start issues. However, it risks overspecialization, often recommending movies too similar to those already watched, thereby limiting diversity in suggestions.

3. Hybrid Recommendation Models Recognizing the limitations of standalone methods, Adomavicius and Tuzhilin (2005) proposed hybrid systems that combine collaborative and content-based approaches. Such systems benefit from the strengths of both models—CF's ability to capture hidden similarities among users and CBF's reliance on item features—while mitigating their weaknesses. These models generally achieve higher accuracy and coverage, making them a widely adopted strategy in industry-grade platforms.

4. Matrix Factorization Techniques. A major breakthrough came with the Netflix Prize Challenge (2009), where Koren et al. demonstrated the effectiveness of matrix factorization methods such as Singular Value Decomposition (SVD). These models decompose the user-movie rating matrix into latent factors that capture underlying patterns (e.g., a user's preference for action over romance). Matrix factorization significantly improved prediction accuracy, scalability, and became the backbone of many large-scale recommender systems. Nonetheless, it still struggles with interpretability and limited adaptability to non-numeric user input.

5. Association Rule Mining association rule mining has also been applied in movie recommendation. This technique uncovers relationships between movies that are frequently co-watched, enabling "if you watched X, you might also like Y" type recommendations. While computationally efficient, these systems lack deep personalization, as they focus on general patterns rather than user-specific intent.

6. Deep Learning Approaches with the surge of deep learning, researchers have developed models that incorporate Convolutional Neural Networks (CNNs) for visual feature extraction (e.g., analyzing movie posters), Recurrent Neural Networks (RNNs) for sequential viewing patterns, and transformer architectures for semantic understanding of textual data such as reviews and synopses. These approaches capture complex, non-linear relationships and enhance recommendation accuracy. However, they require large datasets, high computational resources, and often function as "black boxes," limiting transparency in recommendations.

7. Large Language Model (LLM)-based Recommendation

The most recent development is the integration of Large Language Models (LLMs) and advanced Natural Language Processing (NLP) techniques. Unlike previous systems that rely primarily on structured inputs (ratings, metadata, or embeddings), LLMs can interpret free-text queries directly. For example, a user may express a preference as *"Recommend me a heart-warming family movie with a strong message, similar to 'The Pursuit of Happiness'."* Traditional models struggle to process such queries, whereas LLMs

can extract intent, sentiment, and contextual preferences with **semantic precision. This makes LLM-based** recommendation systems the logical next step, as they align with the natural way users express themselves and enable personalized, flexible, and conversational interaction.

3. Methodology

The methodology of NextFlick involves multiple stages:

1. Data Collection: Extract IMDB datasets for movie metadata and ratings.
2. Data Preprocessing: Clean, normalize, and structure the data.
3. Query Processing: LLMs parse natural language queries into structured intents (e.g., genre, time frame, rating threshold).
4. Feature Extraction: Generate embeddings and vectors for each movie based on metadata and textual descriptions.
5. Recommendation Engine: Hybrid filtering approach combining collaborative and content-based techniques.
6. Ranking: Prioritize recommendations based on relevance, ratings, and user sentiment analysis.
7. Evaluation: Assess system performance using metrics such as precision, recall, and RMSE.

This methodology ensures accuracy, scalability, and adaptability.

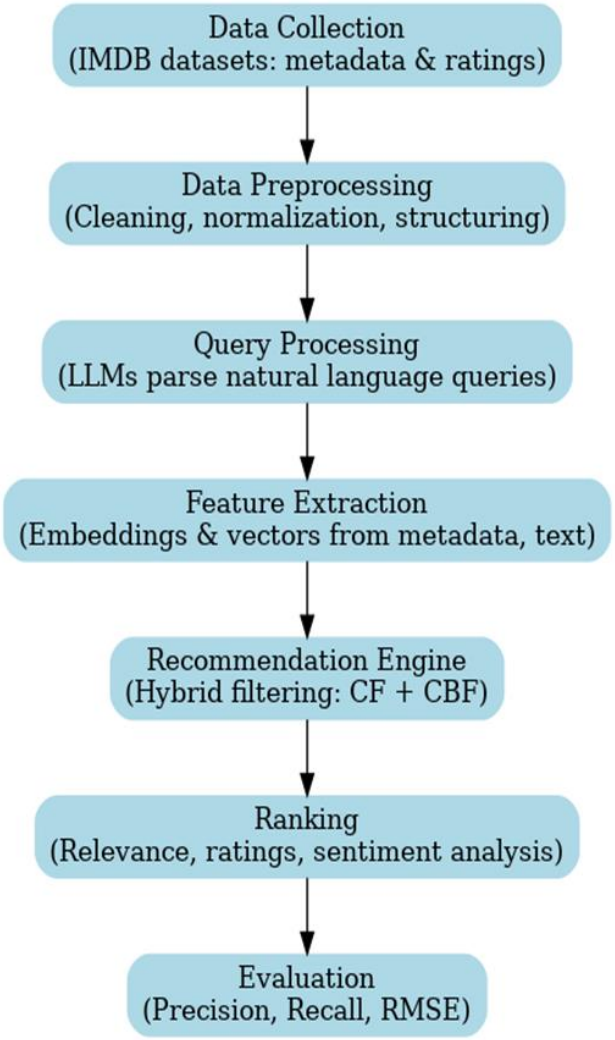


Figure: Flowchart of the NextFlick methodology showing all stages from data collection to evaluation.

4. Algorithms & Techniques

-Collaborative Filtering: Finds similarity among users and items.

1. Collaborative Filtering (CF)

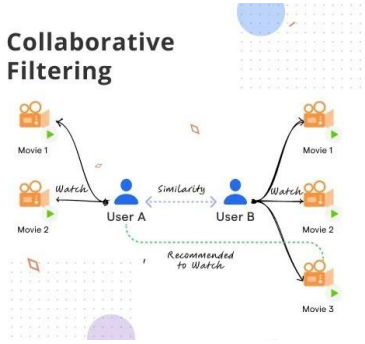
- **User-based CF:** Predict a rating of user *u* for movie *i* using ratings from similar users:

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{v \in N(u)} w(u,v) \cdot (r_{v,i} - \bar{r}_v)}{\sum_{v \in N(u)} |w(u,v)|}$$

Where:

- $\hat{r}_{u,i}$ = predicted rating
- $\bar{r}_u$ = average rating of user *u*
- $w(u,v)$ = similarity between users *u* and *v* (e.g., cosine or Pearson correlation)
- $N(u)$ = set of neighboring users

⚙ Uses similarity functions to measure how close users are.



-Content-Based Filtering: Matches movies based on metadata attributes.

2. Content-Based Filtering (CBF)

- Each movie is represented as a **feature vector**:

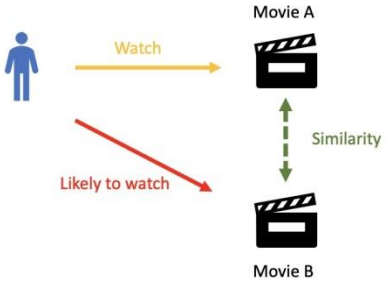
$$\vec{m_i} = (f_1, f_2, f_3, \ldots, f_n)$$

For example, features could be **genre weights, director embeddings, keywords, etc.**

- Prediction is based on similarity between the user profile $\vec{u}$ (built from movies they liked) and the movie vector $\vec{m_i}$:

$$\text{score}(u, m_i) = \cos(\vec{u}, \vec{m_i}) = \frac{\vec{u} \cdot \vec{m_i}}{\|\vec{u}\|\|\vec{m_i}\|}$$

✅ Ensures personalization even for new users.



-Hybrid Filtering: Combines collaborative and content-based to improve accuracy.

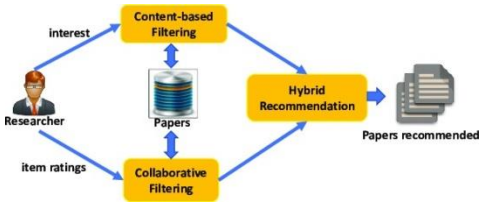
3. Hybrid Filtering

- Linear combination of CF and CBF scores:

$$\text{HybridScore}(u, m) = \alpha \cdot \text{CF}(u, m) + (1 - \alpha) \cdot \text{CBF}(u, m)$$

Where $0 \leq \alpha \leq 1$ is a tunable parameter.

✅ Provides balance between user similarity and movie metadata.



-Cosine Similarity: Measures angle-based similarity between movie vectors.

4. Cosine Similarity

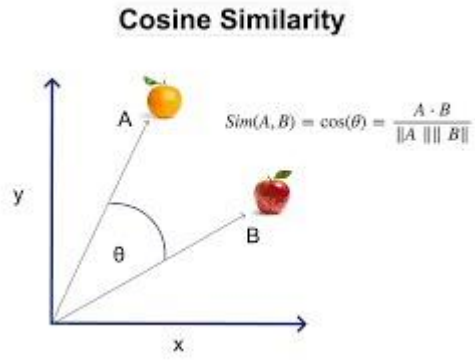
- For two vectors A and B:

$$\cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

Where:

- $A \cdot B = \sum_{i=1}^{n} A_i B_i$ (dot product)
- $\|A\| = \sqrt{\sum_{i=1}^{n} A_i^2}$ (magnitude)

📌 Example: If Movie1 = (1,0,1,0), Movie2 = (1,1,0,0), their similarity = 0.5.



-Matrix Factorization: Factorizes user-item rating matrices to discover latent features.

5. Matrix Factorization (SVD)

- Given a user-item matrix R, decompose into latent factors:

$$R \approx U \cdot \Sigma \cdot V^T$$

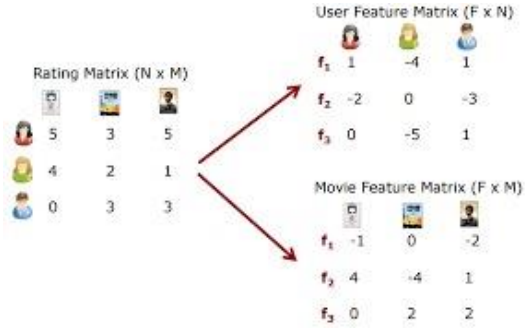
Where:

- U = user-feature matrix
- Σ = diagonal matrix of latent factors
- V = movie-feature matrix
- Prediction:

$$\hat{r}_{u,i} = p_u^T q_i$$

Where p_u is the latent vector for user u and q_i for movie i.

✅ Captures hidden patterns like "user prefers sci-fi thrillers."



-K-Means Clustering: Groups movies into genre clusters for recommendation diversity.

6. K-Means Clustering

- Objective: partition movies into k clusters by minimizing distance from cluster centers:

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$$

Where:

- C_i = cluster i
- μ_i = centroid of cluster i

📌 Example: All superhero movies might cluster together, enabling group-based recommendations.

-Association Rule Mining: Identifies co-watching patterns for rule-based suggestions.

7. Association Rule Mining (Apriori Algorithm)

- Rules like $X \Rightarrow Y$ (if a user watches X, they likely watch Y).

Metrics:

- **Support**: Probability of both X and Y occurring:

$$\text{Support}(X \Rightarrow Y) = \frac{\text{Transactions containing } X \cup Y}{\text{Total Transactions}}$$

- **Confidence**: Likelihood of Y given X:

$$\text{Confidence}(X \Rightarrow Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)}$$

- **Lift**: Strength of rule beyond random chance:

$$\text{Lift}(X \Rightarrow Y) = \frac{\text{Confidence}(X \Rightarrow Y)}{\text{Support}(Y)}$$

✅ Example: "70% of users who watched Iron Man also watched Avengers."

-LLM Query Parsing: Converts free-text into structured recommendation filters.

8. LLM Query Parsing

- Free-text query → Structured filters.
- Example: *"Show me recent comedy movies with high IMDB rating"*.

Steps:

1. **Tokenization** → Break query into words.
2. **NER (Named Entity Recognition)** → Extract entities (genre = comedy, time frame = recent, rating ≥ 8)
3. **Embedding-based similarity**: Represent query Q and movie descriptions M_i as vectors using transformer embeddings:

$$\text{Similarity}(Q, M_i) = \frac{Q \cdot M_i}{\|Q\| \|M_i\|}$$

✅ Allows natural, conversational movie recommendations.

5. System Design & Architecture

The architecture of NextFlick is composed of the following modules:

-User Interface: Accepts free-text queries in natural language.

-LLM Query Parser: Extracts structured information such as genres, release year, and mood.

-Recommendation Engine: Combines collaborative, content-based, and hybrid filtering.

-Ranking Module: Sorts results by similarity, ratings, and query relevance.

-Output Layer: Presents explainable recommendations to the user.

[A block diagram showing User Query → LLM Parser → Recommendation Engine → Ranking → Output Recommendations]

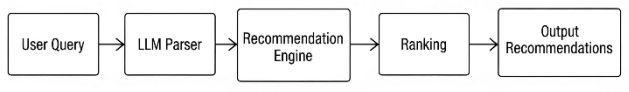


Fig: Architecture of NextFlick System.

6. Evaluation Metrics

To validate system accuracy, several evaluation metrics are used:

1. Precision and Recall

- Precision measures how many of the recommended movies are actually relevant to the user.
- Recall measures how many of the relevant movies were successfully recommended out of all possible relevant ones.
- These two metrics together provide insight into the accuracy and completeness of the system.

📌 Example: If the system recommends 10 movies and 7 are relevant, precision = 0.7. If there were 14 relevant movies in total, recall = 0.5.

2. F1-Score

- Since precision and recall often trade off against each other, the F1-score is used to find a balanced measure.
- Formula:
- A higher F1-score indicates the system is performing well both in terms of precision and recall.

3. RMSE (Root Mean Square Error)

- Evaluates how close the system's predicted ratings are to the actual user ratings.
- Formula:
- Lower RMSE indicates better predictive accuracy of rating values.

4. Diversity and Novelty

- Ensures recommendations are varied across genres, actors, and themes rather than repeatedly suggesting similar items.
- Diversity can be quantified by measuring dissimilarity between recommended items using cosine similarity or Jaccard index.

📌 Benefit: Prevents overspecialization and keeps users engaged by offering broader choices.

- Measures how often the system recommends lesser-known or unexpected movies instead of only popular ones.
- Novelty can be evaluated using popularity-based metrics, where recommendations from the "long tail" are rewarded.

📌 Benefit: Increases discovery of hidden gems and improves user satisfaction.

5. Coverage

- Refers to the proportion of movies in the catalogue that the system is capable of recommending.
- High coverage ensures that the system does not restrict itself to a small subset of the database.

6. User Satisfaction (Subjective Evaluation)

- Beyond technical metrics, user surveys, ratings, and feedback are collected.

- Practical usability tests measure:
  - Ease of interaction with the system.
  - Perceived relevance of recommendations.
  - Overall entertainment value.

📌 Benefit: Provides real-world validation of system effectiveness.

7. Serendipity
   - Measures how often the system surprises the user with unexpected yet enjoyable recommendations.
   - Ensures that suggestions are not only relevant but also engaging.

8. Applications

Applications of NextFlick extend across:
-OTT Platforms: Improve user engagement with precise recommendations.
-AI Assistants: Voice-based recommendations for Alexa, Google Assistant.
-Film Industry: Data-driven insights for targeted promotions.
-Academia: Research on user behaviour and recommendation systems.
-Custom Mobile Apps: Personalized movie apps for niche markets.

1. Personalized Movie Recommendations
   - Provides tailored suggestions based on mood, genre, actors, or specific user queries.
   - Example: A user types *"Show me inspiring movies from the 90s starring Tom Hanks"*, and NextFlick delivers accurate results.
2. Streaming Platforms Integration
   - Can be integrated into OTT platforms (Netflix, Amazon Prime, Disney+, etc.) to improve user retention.
   - Helps platforms stand out by offering natural language–driven discovery.
3. Virtual Assistants & Chatbots
   - Works with AI assistants (Alexa, Google Assistant, Siri) to give real-time movie suggestions via voice commands.
   - Example: *"Suggest me a light-hearted family comedy for tonight."*
4. Cinema & Theatres
   - Useful for cinema websites to recommend shows based on user interests.
   - Can guide audiences toward similar movies currently in theatres.
5. Educational Applications
   - Helps film students or researchers find movies by theme, historical era, or cultural background.
   - Example: *"Show me war documentaries from World War II."*
6. Content Marketing & Advertising
   - Platforms can target users with personalized promotions (e.g., new sci-fi releases for sci-fi fans).
   - Increases engagement and conversions.
7. Social Media Platforms
   - Integration into apps like Facebook, Instagram, or Twitter/X to recommend movies based on trending discussions or user interests.
8. Travel & Lifestyle Apps
   - Apps can suggest movies matching user context (e.g., *"short movies for flights"* or *"romantic movies for honeymoon trips"*).
9. Mental Health & Therapy
   - Curates recommendations based on mood and emotions (e.g., uplifting movies for stress relief).
   - Supports therapeutic entertainment.
10. Cross-Domain Applications
    - Can extend to TV shows, web series, documentaries, anime, and even music videos by adapting the same methodology.

9. Future Scope

1. **Context-Aware Recommendations**: Incorporating contextual factors such as time of day, geographical location, mood, or even current events could enhance personalization. For instance, a user might prefer light-hearted comedies during weekends and short thrillers during weekdays. Integrating contextual signals will make recommendations more dynamic and situationally relevant.

2. **Multimodal Inputs**: Beyond text-based queries, enabling users to interact through voice commands, emojis, or even images can make the system more engaging and inclusive. This would allow for richer query interpretation, especially for younger audiences or those who prefer conversational interfaces like voice assistants.

3. **Cross-Domain Recommendation Systems**: Expanding beyond movies to include TV shows, music, podcasts, and books could transform NextFlick into a comprehensive digital entertainment hub. Cross-domain recommendations also enable deeper personalization by linking user preferences across multiple media types.

4. **Explainable AI (XAI)**: One of the major criticisms of black-box recommendation systems is the lack of transparency. By adopting explainable AI techniques, NextFlick can provide

clear justifications for why a particular movie was recommended (e.g., "Because you liked *Inception*, we recommend *Shutter Island*"). This increases user trust and satisfaction.

5. **Federated Learning and Privacy-Preserving Techniques**: As concerns about data privacy continue to grow, leveraging federated learning allows models to be trained across distributed devices without directly accessing user data. This approach ensures compliance with privacy regulations while still maintaining strong personalization capabilities.

Key Challenges

6. **Cold Start Problem**: New users with little interaction history or newly released movies with limited ratings make it difficult for traditional models to provide accurate recommendations.

7**. Data Sparsity:** Many users only rate a small fraction of the available movies, leading to sparse rating matrices. This reduces the effectiveness of collaborative filtering and weakens recommendation accuracy.

8. **Scalability:** With ever-growing datasets such as IMDB, Netflix, or Amazon Prime catalogues, ensuring real-time recommendations requires advanced optimization and distributed computing strategies.

9. **Overfitting**: Sophisticated machine learning and deep learning models may perform well on training data but fail to generalize to new users or unseen movies. Preventing overfitting requires regularization, dropout, and careful validation strategies.

10**. User Bias and Popularity Dominance:** Popular or trending movies tend to dominate recommendations, overshadowing niche or lesser-known films. This leads to reduced diversity and novelty, diminishing the exploration aspect of recommendations.

11. **Privacy Concerns:** Storing and analysing sensitive user histories (viewing patterns, personal preferences) raises privacy issues. Ensuring compliance with GDPR, CCPA, and other data protection regulations is critical.

10. Conclusion

NextFlick introduces a cutting-edge movie recommendation system that merges the interpretability of LLMs with the proven efficiency of recommendation algorithms. Unlike traditional systems, it supports natural language queries, offering a seamless and personalized movie discovery experience.

By leveraging IMDB datasets, NLP, and hybrid filtering, NextFlick improves accuracy, scalability, and user engagement. With further innovations in context-awareness and explainable AI, it promises to redefine digital entertainment experiences.

11. References

[1]Breese et al., "Collaborative Filtering for Automated Recommendation," 1998.

[2]Panniello et al., "Content-Based Recommender Systems," 2014.

[3]Adomavicius & Tuzhilin, "Hybrid Recommendation Methods," 2005.

[4]Koren et al., "Matrix Factorization Techniques for Recommender Systems," 2009.

[5]Jannach et al., "Cosine Similarity Measures for Recommender Systems," 2015.

[6]Sun et al., "Hybrid Collaborative Filtering Algorithms," 2018.

[7]Bobadilla et al., "Recommender Systems Survey," 2013.

[8]Hill et al., "Recommending and Evaluating Choices in a Virtual Community of Use," 1995.

[9] Katarya R., Verma O.P. An effective collaborative movie recommender system with cuckoo search. Egypt. Inform. J. 2017.

[10] Dakhel G.M., Mahdavi M. A new collaborative filtering algorithm using K-means clustering and neighbors' voting; Proceedings of the 11th International Conference on Hybrid Intelligent Systems (HIS); Malacca, Malaysia. 5–8 December 2011.

[11] Kumar B., Sharma N. Approaches, Issues and Challenges in Recommender Systems: A Systematic Review. Indian J. Sci. Technol. 2016