"Emotion-based Filtering"

Allow users to filter movies or series based on the emotions they want to experience. For instance, categories like "Feel-Good," "Adrenaline Rush," "Mind-Bending," "Heartwarming," or "Tearjerkers."

This could be powered by user reviews tagging specific emotions and a backend system that analyzes the general sentiment of existing reviews. It creates a personalized and mood-driven way to explore content.

1. Collect Emotional Tags from Users

- **User Reviews Integration**: When users submit reviews, allow them to tag their emotional response to the movie/series. Provide predefined options like:
    - "Exciting"
    - "Heartwarming"
    - "Suspenseful"
    - "Inspiring"
    - "Emotional/Touching"
    - "Hilarious"
    - "Mind-Bending"
    - "Sad"
- **Vote for Emotional Tags**: Allow users to vote for emotions they felt after watching. For example, "Does this movie feel thrilling?" (Yes/No).

---

2. Backend Sentiment Analysis

- **Natural Language Processing (NLP)**: Implement sentiment analysis using libraries like Python's NLTK or Google's Natural Language API. Analyze review text to identify keywords and classify the predominant emotion.
- **Tag Weightage System**: Assign a weight to user-selected tags and NLP-identified emotions to determine the dominant emotional category for each movie/series.

3. UI/UX Implementation

- **Filter by Emotion**: Add a filter on the home screen or search bar with emotion categories. Users can select one or multiple emotions to get recommendations.
- **Emotion Highlights**: On each movie/series page, showcase a percentage-based breakdown of emotions. For example:
  - 60% Heartwarming
  - 25% Thrilling
  - 15% Sad
- **Personalized Recommendations**: Suggest movies based on previously tagged emotions of titles users rated or liked.

---

4. Data Enhancement Over Time

- **Crowd-Sourced Refinement**: As more users review and vote on emotions, refine and improve the accuracy of emotion tags.
- **Machine Learning Models**: Train a custom machine learning model to predict emotions for less-reviewed movies/series by analyzing similar titles.

---

5. Incremental Rollout

- Start with manual tagging: For a small database of movies, manually assign emotions based on critical reviews.
- Gradually implement automated systems for larger datasets.
- Roll out emotion-based filters to a limited user base for feedback before full-scale implementation.

Tools and Technologies

- **Frontend**: React Native for seamless UI integration.
- **Backend**: Python or Node.js for NLP and data processing.

- **Database:** Use a relational database like PostgreSQL to store emotional metadata for movies.
- **APIs**:
    - Google Natural Language API for initial NLP processing.
    - Custom API for serving emotion-filtered recommendations.

Next Steps

- Start by designing a basic tagging and filtering system for user reviews.
- Build a prototype for manual emotion-based filtering to test with a small audience.
- Gradually integrate NLP and expand the dataset.