**Movie Recommendation Optimizer**

**Project Goal**

Build an AI-powered recommendation engine and user interface that suggests movies more effectively than traditional browsing. The system will combine **content-based filtering**, **collaborative filtering**, and **hybrid approaches** to deliver accurate and personalized recommendations.

The final deliverable is an **interactive web app** where users can:

1. Select **one or more genres** of interest.
2. Select their **preferred streaming service(s)** (Netflix, HBO, Paramount, etc.).
3. Receive **movie suggestions** that are:
   * Filtered by their choices.
   * Sorted by release year (default), with optional re-sorting by IMDb or Rotten Tomatoes score.
   * Accompanied by an **AI-generated summary** and external review scores (IMDb, Rotten Tomatoes).

This project will be **fun, engaging, and LinkedIn-friendly** while showcasing advanced **data engineering, NLP, ML modeling, UI/UX, and deployment skills**.

**Data Sources**

* **TMDB API** (primary)
  + Metadata: title, genre(s), cast, crew, keywords, release year, popularity, ratings.
  + Endpoints for trending, popular, and detailed movie info.
  + Streaming availability via linked providers (JustWatch integration through TMDB).
* **IMDb / Kaggle Datasets** (secondary)
  + Ratings, tags, reviews.
  + Used to strengthen collaborative filtering and enrich metadata.
* **Rotten Tomatoes API / Web-scraped dataset**
  + Critic score and audience score.
  + Complements IMDb ratings for credibility and external benchmarks.

**Core Features**

**Data Pipeline**

* Fetch, merge, and clean movie metadata from TMDB, IMDb, and Rotten Tomatoes.
* Normalize popularity and ratings across platforms.
* Deduplicate movies, standardize genre tags, and assign streaming service availability.

**Feature Engineering**

* **Genres & Keywords**: One-hot encode or embed multi-genre tags.
* **Overview/Description**: Use TF-IDF, word2vec, or BERT embeddings for semantic understanding.
* **Cast & Crew**: Encode top actors and directors.
* **Ratings & Popularity**: Standardize IMDb, Rotten Tomatoes, and TMDB scores.

**Recommendation Engine**

* **Content-Based Filtering**: Cosine similarity across movie feature vectors.
* **Collaborative Filtering**: Matrix factorization (SVD, Surprise library).
* **Hybrid Model**: Weighted scoring between content and collaborative filtering.
* **Filtering Layer**: Apply genre + streaming service selections before ranking.

**AI Summaries**

* Use GPT-based summarization to generate concise 2–3 sentence descriptions.
* Ensure summaries are standardized in length and tone.

**UI / Frontend**

* **Framework**: Streamlit or Gradio app.
* **Inputs**:
  + Multi-select dropdown for genres.
  + Checkbox or dropdown for streaming platforms.
  + Optional: “Sort by” selector (Year, IMDb, Rotten Tomatoes).
* **Outputs**:
  + List of recommended movies with posters.
  + For each: title, release year, AI summary, IMDb score, Rotten Tomatoes score, and streaming platform tag.

**Evaluation & Metrics**

* **Offline metrics**: Precision@k, Recall@k, RMSE.
* **Online simulation**: Case studies such as “If you liked *Inception*, here are 5 movies you’ll love.”
* **UI validation**: Ensure recommendations adapt correctly when users combine multiple genres/platforms.

**Deliverables**

* **Codebase**: Modular scripts for data collection, preprocessing, feature engineering, model training, and UI deployment.
* **Report (PDF)**: Clear explanation of methods, visuals, results, screenshots of UI, and insights.
* **GitHub Repository**: Professional repo with src/, data/, app/, and report/ directories. Includes deployment instructions.
* **Interactive UI**: Deployed on Streamlit Cloud, Hugging Face Spaces, or Heroku for public demo.
* **LinkedIn Post**: Engaging writeup with visuals and storytelling.

**Success Criteria**

* Fully functional UI where genre + streaming service selection works seamlessly.
* Recommendations sorted and filterable, enriched with summaries and external scores.
* Visualizations that show movie clusters, similarity networks, and keyword clouds.
* Professional GitHub repo and LinkedIn post highlighting both technical depth and creativity.