Project Proposal for Personalized Movie Recommendation System

Capstone Three: Project Proposal
Name: Vidushi Raval
Date: July 21, 2025
Project Title: Personalized Movie Recommendation System
Track: Data Science Career Track
Category: Recommendation Systems, Deep Learning

Problem Identification

Problem Statement -

In today's streaming platforms, users are often overwhelmed by the sheer volume of movie choices.

As a result, they either spend too much time browsing or end up watching content that doesn't align with their preferences. This project aims to solve that problem by building a **personalized movie recommendation system** that can predict a user's movie preferences based on past behavior and movie attributes.

The system will leverage collaborative filtering, content-based filtering, and deep learning to provide more accurate and relevant recommendations.

Context -

Streaming services like Netflix, Amazon Prime, and Hulu thrive on user engagement. The ability to recommend content that users are more likely to enjoy leads to increased watch time, user satisfaction, and subscription retention.

A strong recommendation engine is thus a critical product differentiator. This project emulates the real-world needs of a product team at a streaming company that wants to enhance its recommendation algorithms using machine learning and deep learning.

Criteria for Success

- Generate top-N personalized movie recommendations with high accuracy.
- Achieve strong performance on evaluation metrics such as RMSE (Root Mean Square Error), Precision@K, Recall@K, and NDCG (Normalized Discounted Cumulative Gain).
- Improve recommendation quality using deep learning methods such as Neural Collaborative Filtering.
- Clearly present insights and demonstrate the model's performance through visualizations and interpretability tools.

Scope of Solution Space -

In-scope:

- User-based and item-based collaborative filtering
- Content-based filtering using genres, tags, and metadata
- Matrix factorization techniques (e.g., SVD)
- Neural networks using Keras/PyTorch (e.g., Neural CF, autoencoders)
- Evaluation using ranking and regression metrics
- Visualizations and model explanations

Out-of-scope:

- Real-time streaming data ingestion
- Deployment on a production-scale platform (optional demo app might be created if time allows)

Constraints -

- The dataset is large (20M+ ratings), which could lead to high memory and compute usage.
- The data is anonymized; cold start problems for new users or movies may arise.
- Not all users rate the same number of movies, leading to sparsity.
- Deep learning models may require GPU acceleration for training efficiency.

Stakeholders

- **Primary Stakeholder (hypothetical):** Product team at a streaming service (e.g., Netflix or Amazon Prime)
- **Secondary Stakeholders:**
    - Data science and engineering teams responsible for recommendation infrastructure
    - Marketing teams leveraging recommendations for personalization campaigns
    - End-users who benefit from more relevant content suggestions

Data Sources

Dataset: MovieLens 20M Dataset (Kaggle)
Description:

- 20 million ratings
- 138,000 users
- 27,000+ movies
- Includes: `userId`, `movieId`, `rating`, `timestamp`, `title`, `genres`, `tags`

This dataset will be acquired via direct download from Kaggle, and processed using Pandas and NumPy for exploratory analysis and feature engineering.

Approach (Brief Outline)

1. **Exploratory Data Analysis (EDA):**
   Explore user and movie distributions, rating sparsity, and tag/genre trends.

2. **Baseline Models:**

   Implement collaborative filtering using the **surprise library** to esigned for building and analyzing recommender systems and evaluate it with RMSE(Root Mean Squared Error).

3. **Content-Based Filtering:**
   Use movie genres and tag TF-IDF vectors to compute item similarities.

4. **Deep Learning Models:**
   Train Neural Collaborative Filtering (NCF) and Autoencoder models with Keras or PyTorch to capture non-linear patterns in user-movie interactions.

5. **Model Evaluation:**
   Evaluate models using both regression (RMSE) and ranking metrics (Precision@K, Recall@K, NDCG).

Deliverables

- Code in Jupyter Notebooks (hosted on GitHub)

- Final Capstone PDF Report (submitted via GitHub)

- Slide deck presentation