Final Project Proposal

Movie Recommendations: A Hybrid Recommender System Using Collaborative Filtering, Content-Based Filtering, and Spark-Based Scaling

Authors: Fomba Kassoh, Pricilla Nakyazze

1. Project Title

Movie Recommendations: A Hybrid Recommender System using Collaborative Filtering, Content-Based Filtering, and Spark-Based Evaluation

2. Objective

The goal of this project is to design, implement, and evaluate a robust hybrid movie recommender system that combines Collaborative Filtering (CF) and Content-Based Filtering (CBF) techniques. The system integrates matrix factorization, beyond-accuracy metrics, and Spark-based scalability to simulate a real-world recommender system pipeline.

3. Data Sources

MovieLens 1M Dataset

- 1,000,209 user ratings
- 6,040 users
- 3,883 unique movies
- https://grouplens.org/datasets/movielens/1m

TMDB (The Movie Database) API
Used to enrich MovieLens movies with real metadata

- https://developer.themoviedb.org/docs

TMDB Columns Used

Column	Description
clean_title	Cleaned movie title for API matching
tmdb_id	TMDB movie ID
overview	Plot summary
poster_path	Poster image URL
backdrop_path	Backdrop image URL
vote_average	TMDB average user rating
vote_count	Number of ratings on TMDB
tmdb_genres	Genre names
top_3_cast	Top 3 actors
directors	Director(s)
keywords	TMDB descriptive tags

4. Methodology

Content-Based Filtering (Project 2)

- Features: genres, keywords, top cast, directors (from TMDB)
- Vectorization: TF-IDF and CountVectorizer
- Similarity Metrics: Cosine similarity and Jaccard similarity
- Experimentation: Compare different feature combinations

Collaborative Filtering (Projects 2 and 3)

- Memory-Based: User-User and Item-Item Collaborative Filtering
- Model-Based: SVD (Surprise) and ALS (PySpark MLlib)
- Tuning: Parameters including rank, regularization, and iterations
- Preprocessing: Centering, normalizing, and handling missing values

Hybrid Recommender (Project 3)

- Score blending: Combine CF and CBF scores using weighted average
- Fallback logic: Use CBF for cold-start users or unrated items
- Reranking: Promote diversity and novelty
- Evaluation: Compare hybrid against standalone CF and CBF models

Beyond-Accuracy Evaluation (Project 4)

- Accuracy metrics: RMSE, MAE, Precision@K, Recall@K
- Beyond accuracy:
    - Coverage (catalog utilization)
    - Diversity (genre/cast spread)
    - Novelty (recommendations beyond popular items)
    - Serendipity (pleasant surprises)
- Offline evaluation with a proposal for an online A/B test design

Spark-Based Implementation (Project 5)

- ALS implemented on Apache Spark
- Runtime and memory comparison with in-memory ALS
- Discussion of scaling thresholds for distributed systems
- Platform: PySpark (local mode or Databricks Community Edition)

5. Tools and Technologies

- Languages: Python, PySpark
- Libraries: Pandas, NumPy, Scikit-learn, Surprise, Implicit, Requests, Seaborn, Matplotlib
- Data: ratings.dat, movies.dat, users.dat, movies_enriched_full.csv

- Visualization: Matplotlib, and Seaborn
- Platform: Jupyter Notebook and Databricks

6. Deliverables

- Enriched dataset with TMDB metadata
- Content-Based Filtering recommender with experimental results
- Collaborative Filtering models: ALS, SVD, and memory-based
- Hybrid model with fallback logic and reranking
- Evaluation metrics with accuracy and beyond-accuracy measures
- Spark-based ALS recommender with comparative analysis
- Final report and presentation slides

7. Tasks

- Load MovieLens data, enrich TMDB metadata, and clean
- Build and experiment with Content-Based Filtering
- Build Collaborative Filtering (SVD, UBCF, ALS)
- Combine CF and CBF into Hybrid Recommender
- Evaluate models and implement Spark-based ALS
- Finalize report, code, and presentation

8. Conclusion

This project integrates core concepts from Projects 2 through 5. It covers similarity-based and model-based recommenders, enriches items with external metadata, applies industry metrics beyond accuracy, and concludes with a Spark-based implementation for scalability. The final output demonstrates a complete, production-style recommendation pipeline grounded in academic experimentation and industry relevance.