DSCI 510 – Final Project Progress Report

Student: Rana Shoaaib Mehmood **Project Title:** *Movie Recommender System*

Project Scope Update

The original goal—to develop a personalized movie recommender system using MovieLens user–movie ratings and additional movie metadata—remains unchanged. However, the external data integration plan was refined: instead of using the TMDB API—which was redundant with the Kaggle dataset and limited to retrieving one movie at a time—the project now incorporates IMDb ratings as the third data source, enabling broader coverage, faster access, and a more meaningful basis for external evaluation. The scope now includes comprehensive Exploratory Data Analysis (EDA) and implementation of **Collaborative Filtering** (Matrix Factorization and KNN) methods for recommendation, with IMDb ratings used as a validation benchmark.

Data Sources

1. **MovieLens (ml-latest-small)** – Provides ~100K ratings from 610 users on 9,742 movies. Used for model training and user–item matrix construction.
2. **Kaggle Movies Metadata (TMDB scraped)** – Provides detailed movie attributes such as title, genres, runtime, budget, revenue, and release year. Cleaned, transformed, and merged with MovieLens via tmdbId.
3. **IMDb Ratings Dataset (title.ratings.tsv.gz)** – Supplies average ratings and vote counts for validation of MovieLens-based predictions.

All datasets are accessed programmatically via functions in load.py and processed in process.py. No raw data is committed to the repository.

API / Data Access

The project utilized the Kaggle API to programmatically download the Movies Metadata dataset, which served as the primary source for movie attributes such as genres, runtime, and release year. The TMDB API was also tested during early development but was later discontinued due to redundancy and overlap with the Kaggle dataset, as well as its limitation of retrieving one movie at a time. In addition, the MovieLens ratings and IMDb datasets were programmatically downloaded and integrated using automated data-loading functions to ensure reproducibility and consistency across all data sources. Data retrieval and validation are implemented in reproducible code modules:

- load.py: handles downloading, reading, and cleaning all data sources.
- process.py: performs ID mapping, metadata filtering, and one-hot genre encoding.
- main.py: orchestrates integration and preview.

Issues / Difficulties

- **Data linkage:** Not all MovieLens movies have exact matches in Kaggle metadata; ~2% unmatched.
- **Upcoming tasks:** Implement Matrix Factorization and KNN collaborative filtering, compare predicted ratings against IMDb averages, and finalize visualization/reporting.