ATLAS Internship: Movie Recommender System

From: Jiaqi Zeng

Internship Title: ML/AI

Table of Contents

Basic Information

Client name: ATLAS Internship Program

Name of direct supervisor: Michael Scott Sommers

Names of other team members: No Additional Team Members

Names of contacts who aided during the placement: Megan Fry

Tools utilized during the placement:

- ➢ Python
  - Panda
  - sklearn
- ➢ Flask
- ➢ HTML
- ➢ JavaScript
  - Leaflet.js
- ➢ Dataset
  - TMDB 5000 Movie Dataset
  - The Movies Dataset(with users' ratings)
- ➢ API
  - Movie Showtimes API
  - Movie Poster API
- ➢ GitHub
  - Project Repository

Communication Plan:

- ➢ Weekly Microsoft Team check-ins with direct supervisor (Every Tuesday)
- ➢ Weekly report on Canvas (Every Monday)

Summary of this semester placement expectations:

I was tasked with designing, developing, and deploying a movie recommender web application contains three main sections:

- ➢ **Questionnaire**: Users pick one favorite movie from six popular, high-rated films across diverse genres and indicate why they love it (popularity, story, director, actors, or genre). The system then generates recommendations based on their choices.
- ➢ **Map Section:** Detects nearby cinemas and fetches the next three days of showtimes, allowing users to plan their viewing.
- ➢ **Movie Explore Page:** Enables users to browse movies, provide ratings, and receive collaborative-filtering-based suggestions accordingly.

Project Status Report

Project Overview

- ➢ Project Name: Movie Recommender System
- ➢ Date: May 5. 2025
- ➢ Reporting Period: Spring 2025
- ➢ Prepared by: Jiaqi Zeng

Project Milestones & Status

Milestone	Frontend /Backend	Target Date	Status	Notes

Demographic Filtering with IMDB weighted rating	Backend	Mar 3, 2025	Completed	Visualized top 6 movies globally
Metadata-based Content Filtering	Backend	Mar 10, 2025	Completed	Extracted metadata soup; applied CountVectorizer & cosine similarity
Collaborative Filtering via SVD	Backend	Mar 17, 2025	Completed	Achieved RMSE ≈0.89 [Replaced by a better version in the final project]
Questionnaire & Showtime API Integration	Frontend	Mar 31, 2025	Completed	Integrated questionnaire logic and Google showtime API
Genres-based Structured Similarity	Backend	Apr 7, 2025	Completed	Developed interpretable overlap scoring
Login System & Homepage Layout	Frontend	Apr 14, 2025	Completed	Implemented user auth; organized map and questionnaire sections [Login system is removed in the final project]
Collaborative Filtering(CF) Model Refinement	Backend	Apr 21, 2025	Completed	Binarized ratings matrix; tuned CF and hybrid models
Frontend Cleanup & UI Polish	Frontend	Apr 28, 2025	Completed	Cleaned layout; consolidated recommendation sections
Final Model Polishing & Handover Prep	Both	May 5, 2025	Completed	Prepared final presentation and handover documentation

Project Progress & Achievement

➢ Completed Tasks

- Developed and compared multiple recommendation models: demographic, content-based, and collaborative filtering (user-, item-, and hybrid-based).
- Integrated questionnaire logic to capture user preferences and deliver tailored recommendations.
- Designed and initially implemented a login system for personalized recommendations *(later removed from the final release to simplify user access)*.
- Connected to a third-party API to retrieve cinema showtimes and visualized them on an interactive map.
- Built a full-stack web application with seamless front-end/back-end integration and intuitive user interaction flows.

➢ **Ongoing Tasks**

- **Session-Based Rating Collection:** Implement a mechanism to capture temporary ratings during the current user session and feed them into the collaborative filtering engine for real-time personalization.
- **Cinema Show Synchronization:** Develop logic to match recommended titles with movies currently playing at nearby cinemas, enabling "watch now" suggestions.
- **Questionnaire UX Enhancement:** Redesign the survey flow so users can retake it or explore alternative results without returning to the homepage, improving overall usability.

➢ **Key Achievements**

- Achieved seamless integration of multiple machine-learning models into a unified system.
- Delivered a cohesive web UI that supports interactive recommendations, real-time showtime mapping, and a clear user

journey.

Challenges & Issues

➢ **Data Sparsity (Cold-Start)**

- *Challenge:* New users have few or no ratings, making it difficult to generate accurate recommendations.
- *Mitigation:* Added an onboarding questionnaire to collect initial preference data.

➢ **API Rate Limits**

- *Challenge:* Frequent requests for movie posters and showtimes led to throttling and delays.
- *Mitigation:* Implemented a local caching layer to store recent responses and reduce external calls.

➢ **Recommendation Accuracy**

- *Challenge:* Maintaining high precision@10 and recall across a diverse user base.
- *Mitigation:* Tuned similarity metrics and hybrid-model weights; used RMSE, precision, and recall in an iterative evaluation loop.

Timeline

Week	Date Range	Summary
5	Feb 25 – Mar 03	Implemented demographic filtering using IMDB's weighted rating; visualized top 6 movies among 5,000+ titles.

6	Mar 04 – Mar 10	Built metadata “soup” (actors, directors, genres, keywords); applied CountVectorizer & cosine similarity; started frontend questionnaire UI; planned poster-backend integration & cinema map notification feature.
7	Mar 11 – Mar 17	Implemented collaborative filtering via SVD; achieved RMSE ≈0.89; integrated questionnaire choices; updated frontend to use popular movies as questionnaire options.
8	Mar 25 – Mar 31	Connected Python logic to questionnaire results; researched and tested Google-based showtime API; began integrating cinema showtimes into app.
9	Apr 01 – Apr 07	Developed genres-based recommender scoring by feature overlap; compared with bag-of-words approach; validated recommendations for consistency and interpretability.
10	Apr 08 – Apr 14	Created login system; refined homepage layout (map, showtime modal, questionnaire); deployed recommendation types based on user-selected love reasons.
11	Apr 15 – Apr 21	Built collaborative filtering using ratings & metadata; binarized ratings into liked/disliked; implemented user-, item-, and hybrid filters; compared outputs against actual data.
12	Apr 22 – Apr 28	Refined user-item matrix encoding (1, -1, 0); boosted CF model accuracy; experimented with data sizes & weighting; reorganized and cleaned up frontend layout.
13	Apr 29 – May 05	Finalized model polishing and UI enhancements; drafted handover guide and prepared final presentation materials.

Next Steps

- ➢ **Enhance Collaborative Filtering:** Experiment with matrix factorization methods, refine similarity metrics, and perform systematic hyperparameter tuning to boost recommendation quality.
- ➢ **Add Location-Aware Suggestions:** Leverage user location and viewing history to make recommendations from movies playing at nearby cinemas in real time.

- ➢ **Unify Recommendation Strategies:** Build a hybrid recommender system that blends demographic, content-based, and collaborative approaches for more accurate predictions.
- ➢ **Leverage Implicit Feedback:** Incorporate engagement signals (clicks, saves, watch history) to continuously refine recommendations during a user's session.
- ➢ **Conduct Usability Testing:** Let user to evaluate overall system performance, and guide interface and algorithm improvements.

Personal Summary

This internship has been an incredibly valuable learning experience. I developed practical, hands-on skills across a wide range of areas in machine learning, web development, and problem solving. Working on a full-stack web application allowed me to see how different technologies come together to form a functional product, and how user experience and system logic must be tightly integrated.

One of the most important lessons I learned was the value of self-directed problem solving. I faced several technical challenges such as CORS issues, API integration problems, and session management bugs. Instead of relying on external help, I took initiative to research, debug, and test potential solutions. This process helped me build confidence in navigating unfamiliar technical terrain independently.

On the development side, I gained significant experience building a complete web application. I implemented both the backend in Flask and the frontend using HTML and JavaScript. I also incorporated user login functionality and dynamic features such as cinema map integration and recommendation displays. Working with API endpoints and handling real-time data responses sharpened my ability to build flexible and interactive web systems.

From a machine learning perspective, I explored several recommendation system techniques, including demographic filtering, content-based filtering, and collaborative filtering (user-based, item-based, and hybrid). I also learned to evaluate model performance using metrics like RMSE. Implementing hybrid approaches helped me better understand how to balance accuracy and diversity in recommendations. These insights deepened my understanding of recommender systems and broadened my machine learning skill set.

Overall, this internship strengthened both my technical expertise and my ability to independently solve real-world problems. I feel more prepared and motivated to pursue advanced topics in machine learning and to continue building systems that connect people with information they care about.