**CSC 575 – Self-Evaluation Report**

---

**Evaluation Criteria Reflection**

**1. Code Submission:**

We submitted a complete and functional Jupyter Notebook as our main prototype. The code implements all core features of our movie recommendation system, including data processing, search, ranking, multiple recommendation models, and a user interface.

**2. Code Documentation;**

Our code includes comments explaining key steps and logic. Additionally, we are providing a README.md file with setup instructions, usage guidance, library requirements, and links to public datasets. This makes it easy for anyone to run and test our system.

**3. Basic IIR Techniques:**

We used foundational Information Retrieval techniques like inverted indexing, keyword search using TF-IDF, and faceted filtering based on structured metadata (genres, languages, release year). These formed the backbone of our basic retrieval system.

**4. Advanced IIR Techniques:**

We implemented the **Bayesian Personalized Ranking (BPR)** model using LightFM, which provides personalized, collaborative filtering-based recommendations. Furthermore, we implemented a **hybrid recommendation strategy** that combines popularity-based and BPRbased methods based on user interaction history.

**5. Model Complexity:**

While we used established libraries (e.g., LightFM), we customized the recommendation logic and system design. Our hybrid model was coded manually with a switch mechanism based on the number of ratings per user. We also worked on integrating various models into a single unified workflow.

**6. Dataset Quality:**

We used **The Movies Dataset**, which includes rich metadata such as movie titles, overviews, genres, ratings, cast, and crew. This dataset is much larger and more comprehensive than those used in homework assignments and is well-suited for testing a real-world recommender system.

### 7. Experiments and Evaluation:

We evaluated our models using ranking metrics such as **precision@k** and **recall@k**. We also compared the performance of different recommendation strategies qualitatively and plan to add grid search evaluation for BPR parameters (like number of components).

### 8. Project Objectives:

We closely followed our original proposal. The final system allows users to search, browse, and receive ranked recommendations. All core objectives were achieved:

- Efficient retrieval based on text and metadata
- Personalized recommendations via BPR
- Faceted filtering and intuitive browsing
  We also implemented professor feedback by creating a hybrid model and improving system unification.

**Working parts:**

Took the lead on implementing the **Bayesian Personalized Ranking (BPR)** model using LightFM. He also designed and coded the **hybrid switching logic**, which chooses between BPR and popularity-based models based on user rating history. In addition, I handled much of the debugging and testing, ensuring the system worked smoothly across different user types.

Focused on building the **content-based recommendation system** using TF-IDF across genres, keywords, and overviews. Saami also contributed significantly to the **Gradio-based UI**, helping make the system easy to interact with and user-friendly. He played an important role in testing functionality and helping refine the overall experience.

Worked on implementing the **popularity-based recommendation system** and integrating it into the broader system logic. He also helped with manual testing of cold-start vs. active users, and coordinated efforts across the team. Adnan contributed to drafting the evaluation methods and supported preparing visuals and the final submission files.

Together, we all worked on preprocessing the dataset, experimenting with different recommendation strategies, writing the final report and README, and making sure the final system reflected the goals we started with. This project really taught us how different IIR techniques can come together in a real system.