Movie Recommendation System

1. Project Title

Movie Recommendation System using Content-Based and Collaborative Filtering

2. Objective

To build a smart and interactive movie recommendation system that provides personalized movie suggestions to users using Content-Based Filtering and Collaborative Filtering techniques. The system includes a Streamlit GUI for seamless user interaction.

3. Datasets Used

1. TMDB 5000 Movie Dataset

- Used for Content-Based Filtering
- Contains movie metadata like genres, cast, crew, keywords, and overview
- Helps generate similarity-based recommendations

2. MovieLens Small Latest Dataset

- Used for Collaborative Filtering
- Includes user ratings for various movies
- Useful for modeling user preferences based on interaction patterns

4. Recommendation Approaches

1. Content-Based Filtering

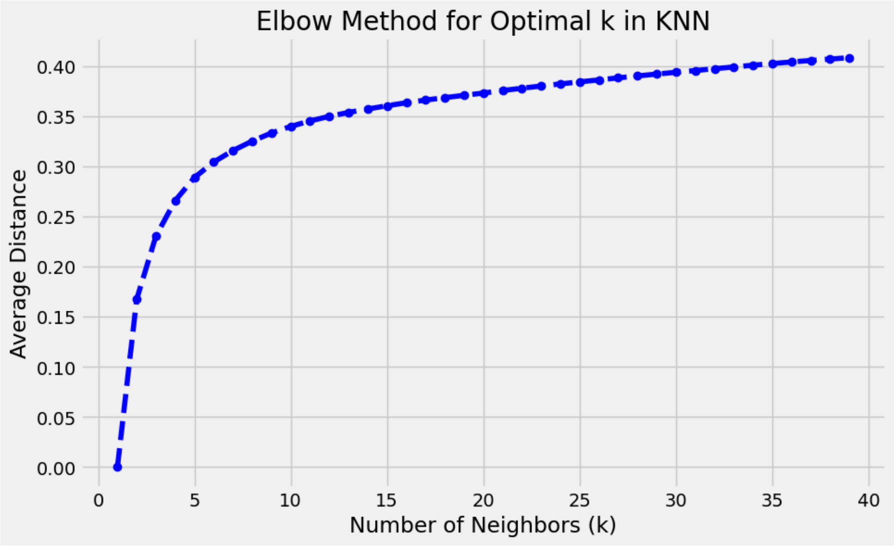
- Features Used: Genre, Director, Cast, Keywords
- Technique: TF-IDF vectorization + Cosine Similarity
- Goal: Recommend movies similar to a selected one based on metadata
- Implementation: /Content based/ folder

2. Collaborative Filtering

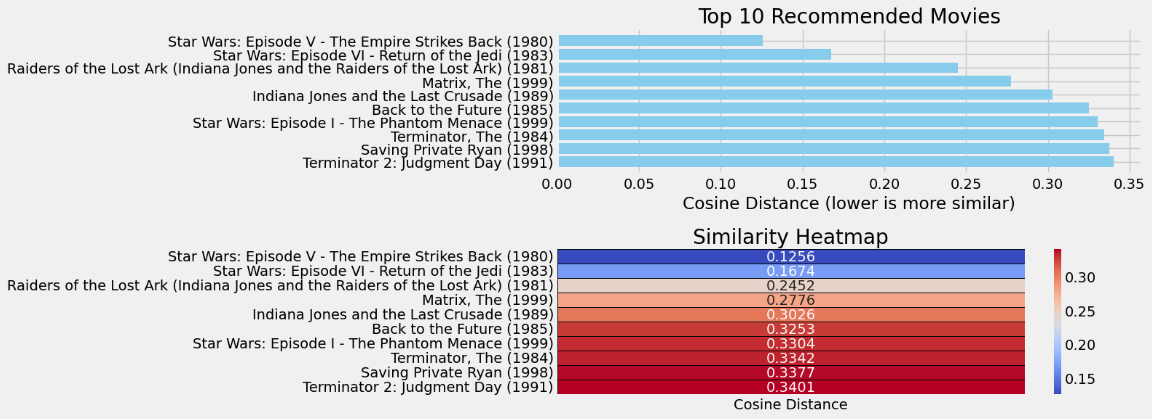
- Approach: K-Nearest Neighbors (KNN) based Collaborative Filtering
- Library Used: sklearn.neighbors.NearestNeighbors
- Technique: Cosine distance between user vectors in the user-item matrix
- Goal: Recommend movies liked by users with similar preferences
- Implementation: /Collaborative Filtering/ folder

5. Performance Metrics

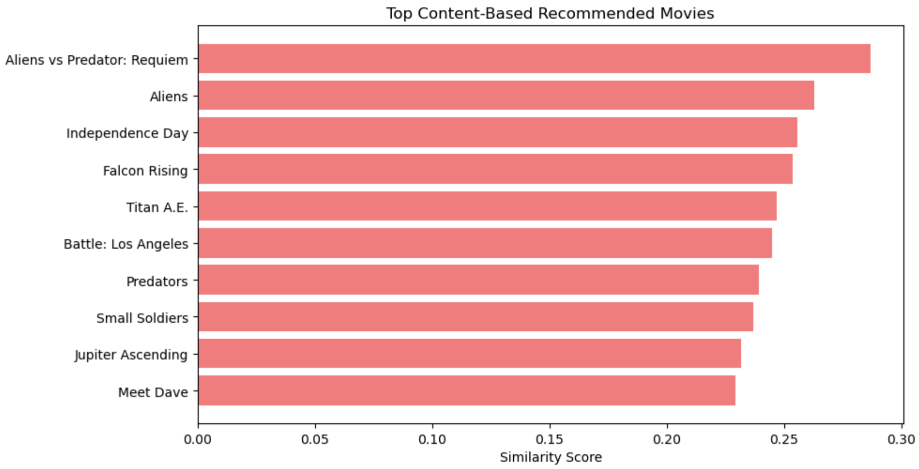
1. Content-Based Filtering

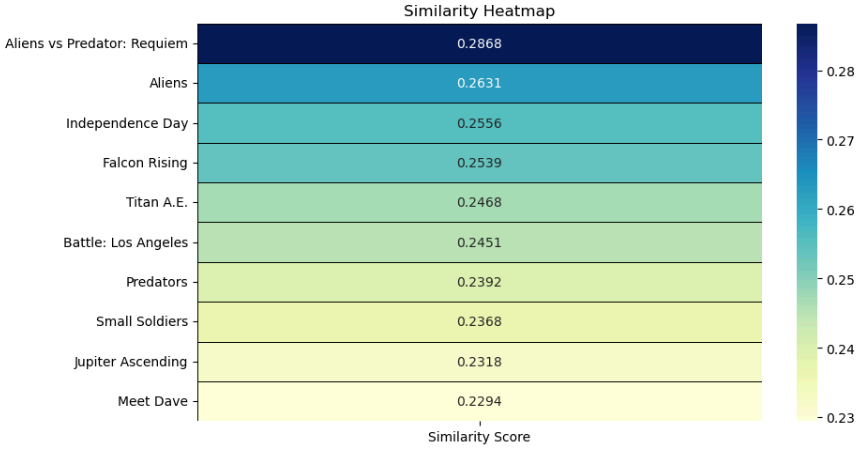


Choosing n_neighbors = 10



2. Collaborative Filtering





5. Tech Stack

- Language: Python 3.x
- Libraries: Pandas, NumPy, scikit-learn, Streamlit
- Development Tool: Jupyter Notebooks
- Interface: Streamlit GUI for interactive movie recommendations

6. Challenges Faced

- Designing an intuitive yet simple GUI for user interaction
- Merging results from two different recommendation approaches
- Handling sparse user-item matrices in collaborative filtering
- Preprocessing and cleaning movie metadata from TMDB

7. Learnings

- Hands-on experience with recommendation algorithms
- Implementation of both content-based and collaborative recommender systems
- Use of sklearn for building custom recommendation logic
- GUI development with Streamlit for quick deployment and real-time testing