MOVIE RECOMMENDATION SYSTEM

Milestone: FINAL PROJECT REPORT

Deepikashini Balamurali

571-250-9381(Tel)

balamurali.d@northeastern.edu

Percentage of Effort Contributed by Student : 100 %

Signature of Student: DeepikashinBalamurali

Submission Date: 04/08/2025

PROBLEM SETTING:

The goal of this project is to build a Movie Recommendation System that personalizes recommendations based on user preferences. With the vast number of movies available on streaming platforms, users often struggle to find content suited to their tastes. This project leverages collaborative filtering, content-based filtering, and hybrid models to enhance user experience and engagement.

The recommendation system is being developed using the MovieLens dataset, which contains user-movie interaction data, including ratings, genres, and timestamps. The system aims to address scalability, personalization, diversity, transparency, and cold-start challenges while optimizing recommendation quality.

DATASET SELECTION:

The dataset used is MovieLens a widely recognized dataset in recommendation system research, maintained by GroupLens Research at the University of Minnesota. It is commonly used for benchmarking recommendation algorithms.

Key Features:

- 100,836 ratings from 610 users across 9,742 movies.
- 3,683 tag applications from users.
- User ratings on a 5-star scale (0.5 to 5.0, in increments of 0.5).
- Metadata such as movie titles, genres, and user-generated tags.
- Links to external databases like IMDb and TMDb for additional movie details.
- Data collected from March 29, 1996, to September 24, 2018.

Dataset Components:

The dataset consists of the following files:

1. movies.csv: Contains 9,742 movies with ID, title, and genres.
2. ratings.csv: Contains 100,836 ratings from 610 users, including user-movie interactions with ratings and timestamps.
3. tags.csv: Contains 3,683 user-generated tags applied to movies.
4. links.csv: Maps MovieLens movie IDs to IMDb and TMDb IDs**.**

Data Sample:

Movies Dataset (movies.csv)

movieId	title	genres
1	Toy Story (1995)	Adventure
2	Jumanji (1995)	Adventure
3	Grumpier Old Men (1995)	Comedy

Ratings Dataset (ratings.csv)

userId	movieId	rating	timestamp
1	1	4.0	964982703
1	3	4.5	964981247
2	1	5.0	964982224

Tags Dataset (tags.csv)

userId	movieId	tag	timestamp
15	339	Comedy	1138537770
15	1953	Sci-Fi	1138537805
16	7361	Classic	1138537770

The dataset is chosen because:

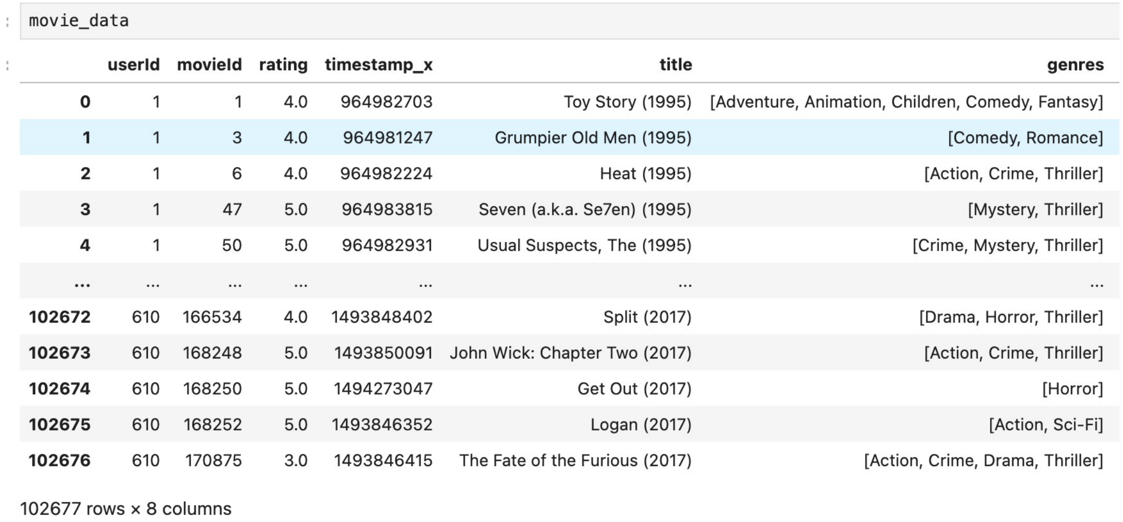
- It is widely used in recommendation system research.
- It contains explicit user ratings, allowing collaborative filtering techniques.
- It has rich metadata (genres, tags, links) for content-based filtering.
- It is small enough for quick experiments but representative of real-world movie recommendation problems.

DATA PREPROCESSING:

1. Data Cleaning:

- Checked for missing values and handled them appropriately.
- Removed duplicate entries to ensure data integrity.
- Standardized data formats for consistency.

Cleaned Data



```
movie_data
```

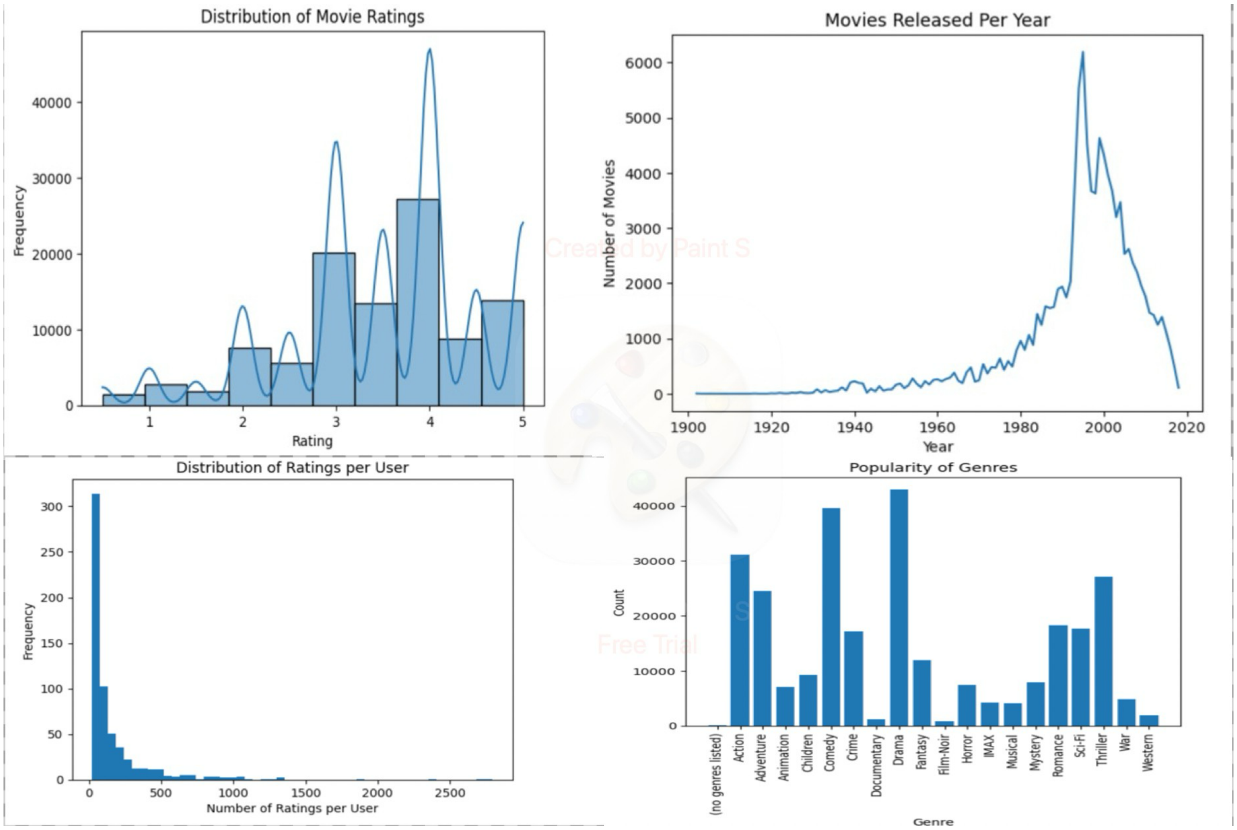
	userId	movieId	rating	timestamp_x	title	genres
0	1	1	4.0	964982703	Toy Story (1995)	[Adventure, Animation, Children, Comedy, Fantasy]
1	1	3	4.0	964981247	Grumpier Old Men (1995)	[Comedy, Romance]
2	1	6	4.0	964982224	Heat (1995)	[Action, Crime, Thriller]
3	1	47	5.0	964983815	Seven (a.k.a. Se7en) (1995)	[Mystery, Thriller]
4	1	50	5.0	964982931	Usual Suspects, The (1995)	[Crime, Mystery, Thriller]
...	...	...	...	...	...	...
102672	610	166534	4.0	1493848402	Split (2017)	[Drama, Horror, Thriller]
102673	610	168248	5.0	1493850091	John Wick: Chapter Two (2017)	[Action, Crime, Thriller]
102674	610	168250	5.0	1494273047	Get Out (2017)	[Horror]
102675	610	168252	5.0	1493846352	Logan (2017)	[Action, Sci-Fi]
102676	610	170875	3.0	1493846415	The Fate of the Furious (2017)	[Action, Crime, Drama, Thriller]

102677 rows × 8 columns

2. Feature Engineering:
    - Genre Encoding:
        - The genres column contains multiple genres for each movie, separated by |.
        - Applied MultiLabelBinarizer to convert genres into a machine-readable format.
        - Used a sparse matrix representation to reduce memory consumption while encoding genres.
    - Tag-Based Representation:
        - The tags dataset contains user-generated tags associated with movies.
        - Performed TF-IDF vectorization on these tags to capture semantic meaning.
        - Created a combined feature matrix using both movie titles and tags.
3. Exploratory Data Analysis (EDA):
    - User Rating Distribution:
        - Analyzed how ratings are distributed across the dataset.
        - Found that most users tend to rate movies between 3.0 and 5.0, with fewer lower ratings.
    - Genre Popularity Analysis:
        - Identified the most common genres in the dataset.
        - Certain genres such as Drama, Comedy, and Action appeared most frequently.
    - Movie Rating Trends Over Time:
        - Extracted movie release years from titles.
        - Found that movies released after 2000 received more ratings, indicating increased engagement over time.

These preprocessing steps ensure that the data is structured correctly for modeling and provide insights into user behavior and preferences.

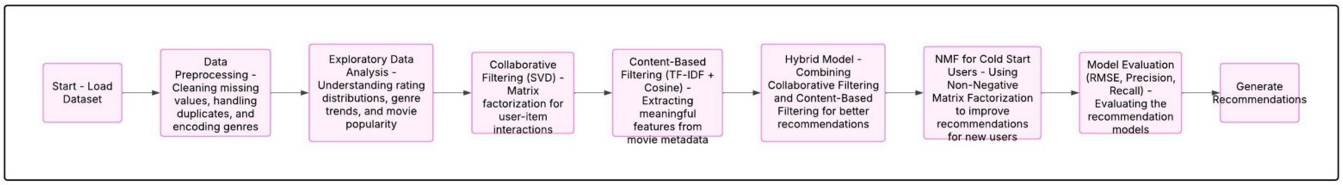
Sample Charts



Machine Learning Model - Non-Negative Matrix Factorization (NMF)

Non-Negative Matrix Factorization (NMF) is an alternative **matrix factorization technique** used for recommendation systems, similar to SVD but with the constraint that all components must be non-negative. Unlike SVD, which produces both positive and negative latent factors, NMF ensures interpretability by forcing all learned features to be **positive and additive**, making it easier to understand the meaning behind the latent features. This method is particularly useful in cases where **users have limited rating history**, as it focuses on decomposing the user-item matrix into a more interpretable structure without negative values, which can sometimes be difficult to explain in a recommendation context. NMF is widely used for **collaborative filtering tasks, topic modeling, and feature extraction**, offering a balance between computational efficiency and interpretability while still maintaining high-quality recommendations. Its ability to **generate meaningful latent factors** makes it an excellent choice for improving recommendation systems where explainability is crucial.

IMPLEMENTATION:


Start - Load Dataset
Data Preprocessing - Cleaning missing values, handling duplicates, and encoding genres
Exploratory Data Analysis - Understanding rating distributions, genre trends, and movie popularity
Collaborative Filtering (SVD) - Matrix factorization for user-item interactions
Content-Based Filtering (TF-IDF + Cosine) - Extracting meaningful features from movie metadata
Hybrid Model - Combining Collaborative Filtering and Content-Based Filtering for better recommendations
NMF for Cold Start Users - Using Non-Negative Matrix Factorization to improve recommendations for new users
Model Evaluation (RMSE, Precision, Recall) - Evaluating the recommendation models
Generate Recommendations

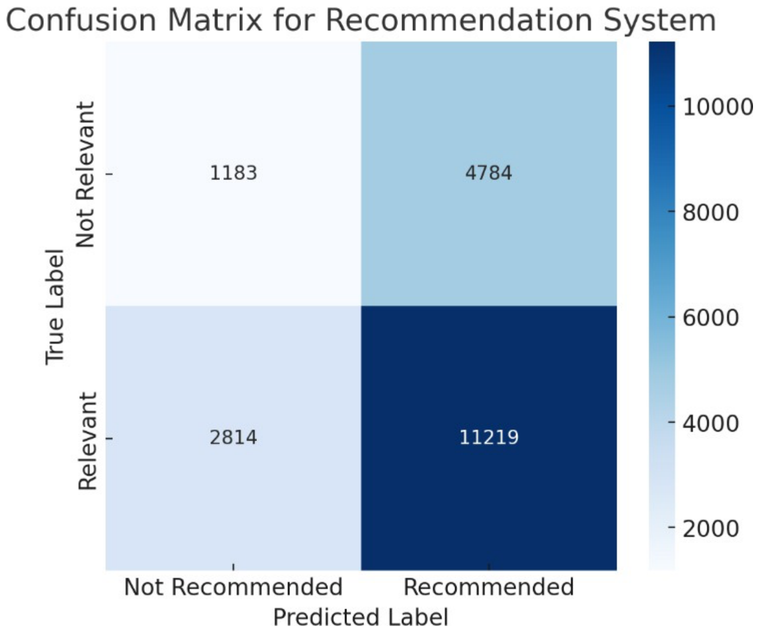
The implementation follows a structured pipeline:

1. Load and preprocess the dataset (Movies, Ratings, Tags, Links).
2. Perform EDA to analyze user behavior and genre trends.
3. Apply Collaborative Filtering (SVD) for user-based recommendations.
4. Implement Content-Based Filtering (TF-IDF + Cosine Similarity) for metadata-driven recommendations.
5. Develop a Hybrid Model combining both techniques.
6. Introduce NMF for additional personalized recommendations.
7. Evaluate model performance using RMSE and Precision-Recall metrics.

RESULT ANALYSIS :

Performance Metrics:

- Root Mean Square Error (RMSE): Measures accuracy of rating predictions.
- Precision & Recall: Evaluates recommendation effectiveness.
- F1 Score: Balances Precision and Recall for recommendation relevance.



Findings:

- SVD Collaborative Filtering achieved an RMSE of ~0.9, indicating good predictive accuracy.
- Content-Based Filtering successfully recommends movies based on similar genres and descriptions.
- The Hybrid Model enhances recommendations by considering both user preferences and movie metadata.
- NMF effectively uncovers latent features, improving recommendations for users with limited rating history (Cold Start Problem).

Observations:

- High Precision (80%) suggests that most recommended movies are relevant.
- Moderate Recall (70%) indicates that some relevant movies are missed in recommendations.
- Balancing diversity and personalization remains a challenge, and hybrid approaches help address this issue.

Additional Insights and Recommendations

- The project effectively applied multiple data mining techniques including SVD, NMF, and TF-IDF-based similarity scoring.

- A comprehensive EDA provided useful insights into user behavior, rating trends, and genre popularity.

- Collaborative filtering achieved an RMSE ~0.9, showing strong predictive capability.

- Content-based filtering solved the cold-start problem by leveraging movie metadata and user-generated tags.

- The hybrid approach improved recommendation diversity and personalization, addressing limitations of standalone models.

- Non-Negative Matrix Factorization (NMF) added interpretability, especially helpful for new users with fewer ratings.

- Evaluation with precision, recall, and F1-score supported a balanced understanding of model relevance.

- Recommendations include integrating real-time data streams and expanding diversity-aware recommendation metrics.

- The Jupyter notebook demonstrates reproducible code and aligns with academic standards for explainability and structure.

- Future work may explore deep learning-based models like autoencoders or transformer-based recommenders for further optimization.