Netflix recommendation System

Student: Mangesh Patil

Seidenberg School of Computer Science & Information Systems

Introduction

Recommendation systems form the backbone of platforms like Netflix, facilitating personalized content discovery and enhancing user engagement. With the proliferation of machine learning and data mining techniques, recommendation systems have evolved to deliver more accurate and relevant recommendations, thereby catering to diverse user preferences and behaviors. These systems leverage historical user interactions and item attributes to predict user preferences and provide tailored recommendations, thereby enhancing user satisfaction and retention.

Literature review

Collaborative Filtering Techniques

Content-Based Filtering Techniques

Hybrid Recommendation Systems

Data Collection

#	Column	Dtype
---	------	-----
0	MovieID	int64
1	title	object
2	genre	object
3	original_language	object
4	overview	object
5	popularity	float64
6	release_date	object
7	vote_average	float64
8	vote_count	int64
9	CustomerID	int64
10	Rating	int64
11	RatingDate	object

I took the Netflix price data set released in 2006 which had customer and movie IDs and ratings

Then I took a dataset from Kaggle containing 1000 movie titles movie ID descriptions, cast, etc merged them, and created a hypothetical data set for performing content-based and collaborative filtering on the same data.

Due to computational restrictions, I took only 5 random ratings from users for each movie title and made my dataset computationally processable,

Materials and methods

Performance of Content-Based Filtering:

TF-IDF Vectorization:

Tokenize the movie descriptions into individual words or tokens.

Convert the tokens into numerical representations using TF-IDF (Term Frequency-Inverse Document Frequency) vectorization.

TF-IDF assigns weights to each word based on its frequency in the document and its rarity across all documents in the dataset. This helps in capturing the importance of words in distinguishing one document from another.

Computed the cosine similarity between each pair of movies based on their TF-IDF vectors.

Cosine similarity measures the cosine of the angle between two vectors and provides a measure of similarity between them. In the context of recommendation systems, it quantifies how similar two movies are based on their textual descriptions.

```
Recommendations for 'The Dark Knight':
1059        Batman: The Long Halloween, Part One
1101        Batman: The Long Halloween, Part Two
688                      The Dark Knight Rises
2410                                    Batman
5010                   Batman: The Killing Joke
342      Batman: The Dark Knight Returns, Part 2
709                                 The Batman
9434                            Batman Forever
655                   Batman: Under the Red Hood
4508                Batman: Gotham by Gaslight
Name: title, dtype: object
```

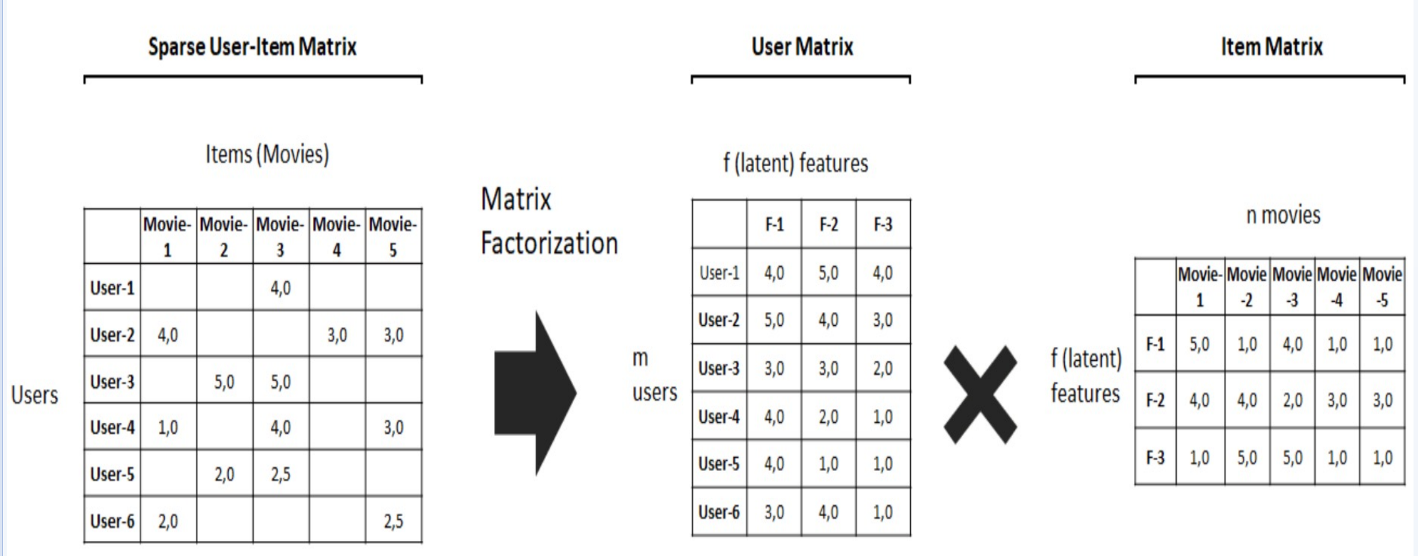
Performance of Collaborative Filtering:

The dataset comprises user ratings for movies, which are formatted using the Surprise library for analysis. The methodology employed involves training and evaluating the SVD algorithm through 5-fold cross-validation, with RMSE and MAE serving as key evaluation metrics. Through iterative training on diverse data subsets, the SVD algorithm generates collaborative recommendations for each user within the test set. The system's performance is gauged by computing the average RMSE and MAE across multiple cross-validation folds.

```
Enter a Movie Title, Movie ID, or CustomerID:  842185
Collaborative Filtering Recommendations for CustomerID 842185
Recommendation 1: Movie ID 5, Estimated Rating 3.5217015557248486
Recommendation 2: Movie ID 6, Estimated Rating 2.9403227586983034
Recommendation 3: Movie ID 11, Estimated Rating 3.4974246481248317
Recommendation 4: Movie ID 12, Estimated Rating 3.713118145679294
Recommendation 5: Movie ID 13, Estimated Rating 3.780180529747466
Recommendation 6: Movie ID 14, Estimated Rating 3.5610813138871187
Recommendation 7: Movie ID 15, Estimated Rating 3.565470606774091
Recommendation 8: Movie ID 16, Estimated Rating 3.35572210272161
Recommendation 9: Movie ID 17, Estimated Rating 3.50410416520919
Recommendation 10: Movie ID 18, Estimated Rating 3.4703688422766024
```

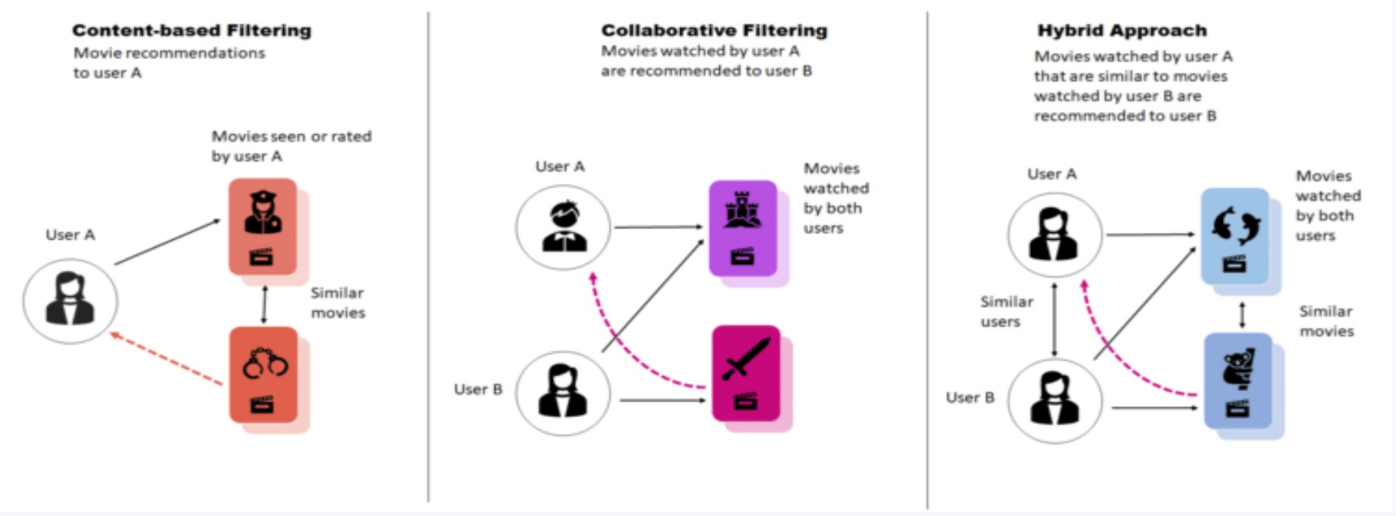
Dimensionality reduction

One of the most widely used techniques for dimensionality reduction is matrix factorization. This approach compresses the initial sparse user/item matrix and presents it as separate matrices that present items and users as unknown feature vectors Such a matrix is densely populated and thus easier to handle, but it also enables the model to uncover latent dependencies among items and users, which increases model accuracy



Hybrid Recommendation System:

By amalgamating recommendations from both collaborative and content-based filtering approaches, our hybrid system capitalizes on the unique strengths of each method. Combining top-rated movies from collaborative filtering with thematically similar recommendations from content-based filtering, our hybrid approach delivers a diverse array of personalized suggestions. This synthesis of methodologies enables us to cater to a broader spectrum of user preferences, resulting in heightened satisfaction and engagement levels.



Result and Analysis

The cold start problem poses significant obstacles for recommendation systems, as they rely on historical data to make accurate predictions and provide relevant recommendations. Without sufficient data about users or items, the system may struggle to deliver personalized experiences, leading to potential user dissatisfaction and reduced effectiveness of the recommendation system.

To address it we have different ways, one of which is our hybrid recommendation system. Combining collaborative filtering with content-based methods to mitigate the cold start problem and improve recommendation quality for new users and items.

In summary, our comprehensive evaluation affirms that our recommendation system adeptly leverages collaborative and content-based filtering techniques to deliver accurate, diverse, and personalized recommendations. These findings not only validate the efficacy of our approach but also underscore its pivotal role in enhancing the user experience and driving positive outcomes for our platform or service

References

•Bennett, J., & Lanning, S. (2007). The Netflix prize. In KDD Cup and Workshop in conjunction with KDD (Vol. 2007, p. 35).

•Zhao, Y., Shang, J., & Qiu, M. (2020). Deep reinforcement learning for recommendation with implicit feedback. Information Sciences, 541, 1-14.

•McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & Agüera y Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data. In Artificial Intelligence and Statistics (pp. 1273-1282).

•Kleinberg, J., Ludwig, J., Mullainathan, S., & Obermeyer, Z. (2018). Algorithmic fairness. AEA Papers and Proceedings, 108, 22-27.