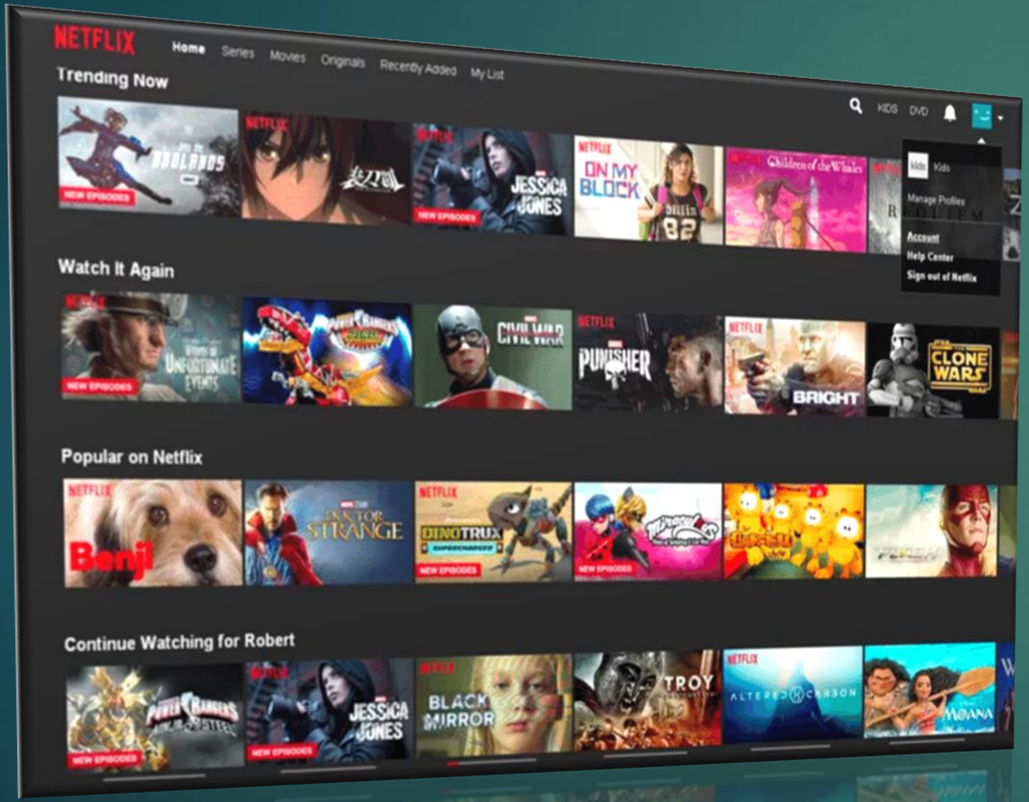
Question: How can we effectively predict user preferences for movies they haven't seen based on their past ratings?

EBRAHIM GOLRIZ

The Challenge of Information Overload



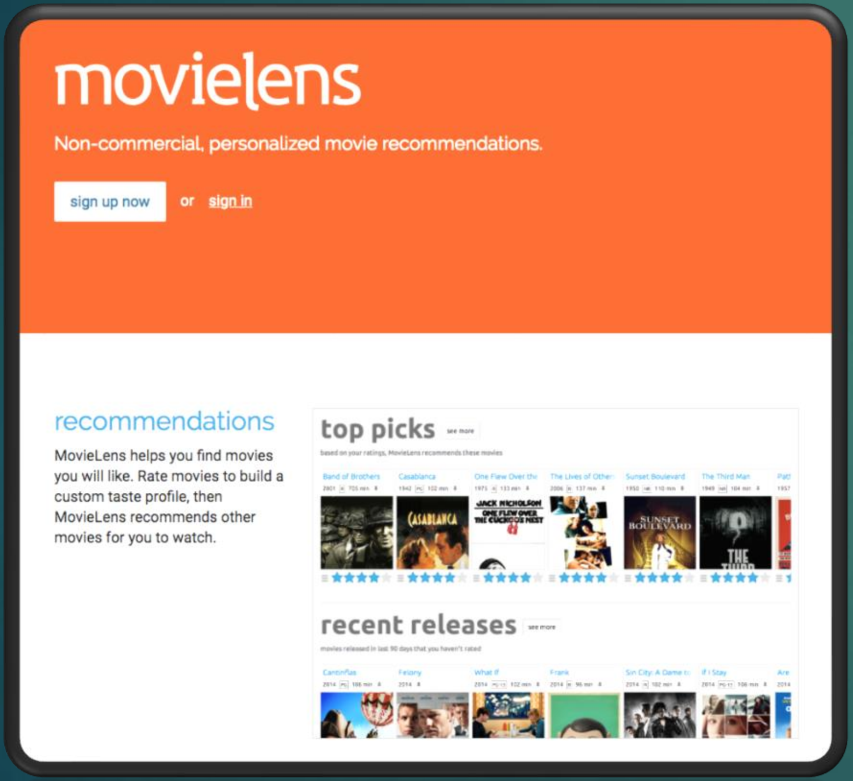
- Explosion of **content** in the digital age (movies, products, etc.). Users need help finding what they like.
- Recommending products that users want, will increase the **profit** and number of **sales**

Recommender Systems as a Solution

- Personalized experiences
- Improved user engagement
- Business benefits (e.g., increased sales, user retention)

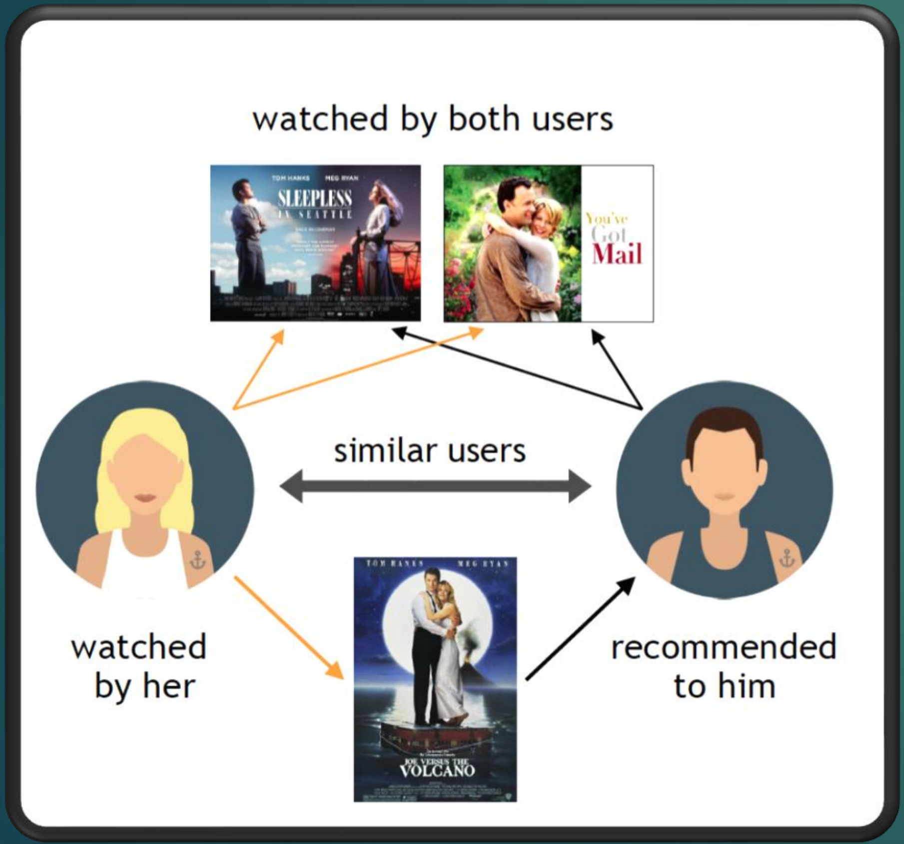
Dataset

Movielens is a well-known, publicly available dataset



- **Ratings**
  - UserID, MovieID, Rating(1-5 Stars)
- **Movies**
  - Title, MovieID, Genres
- **Tags**
  - UserID, MovieID, Tags

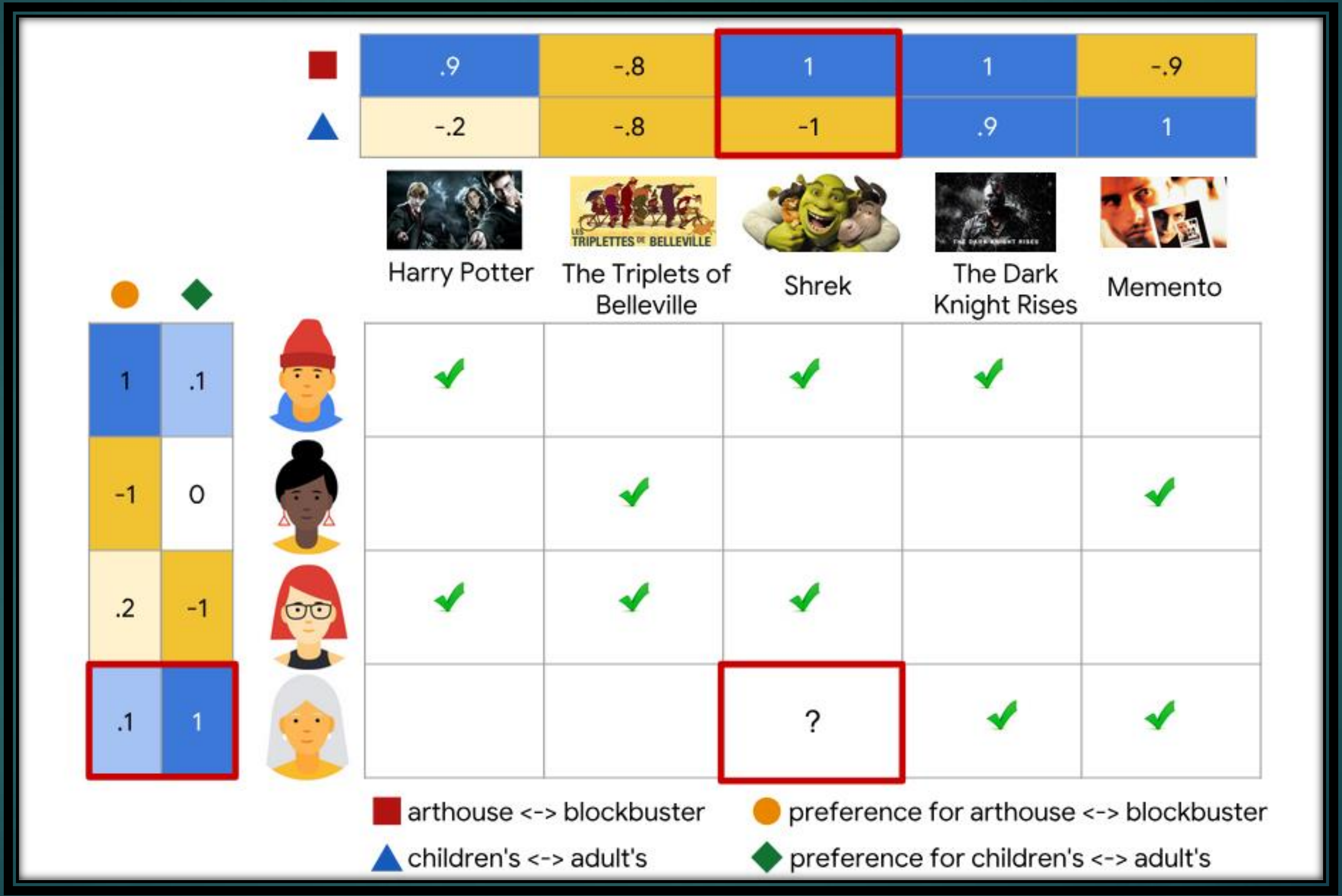
Collaborative Filtering



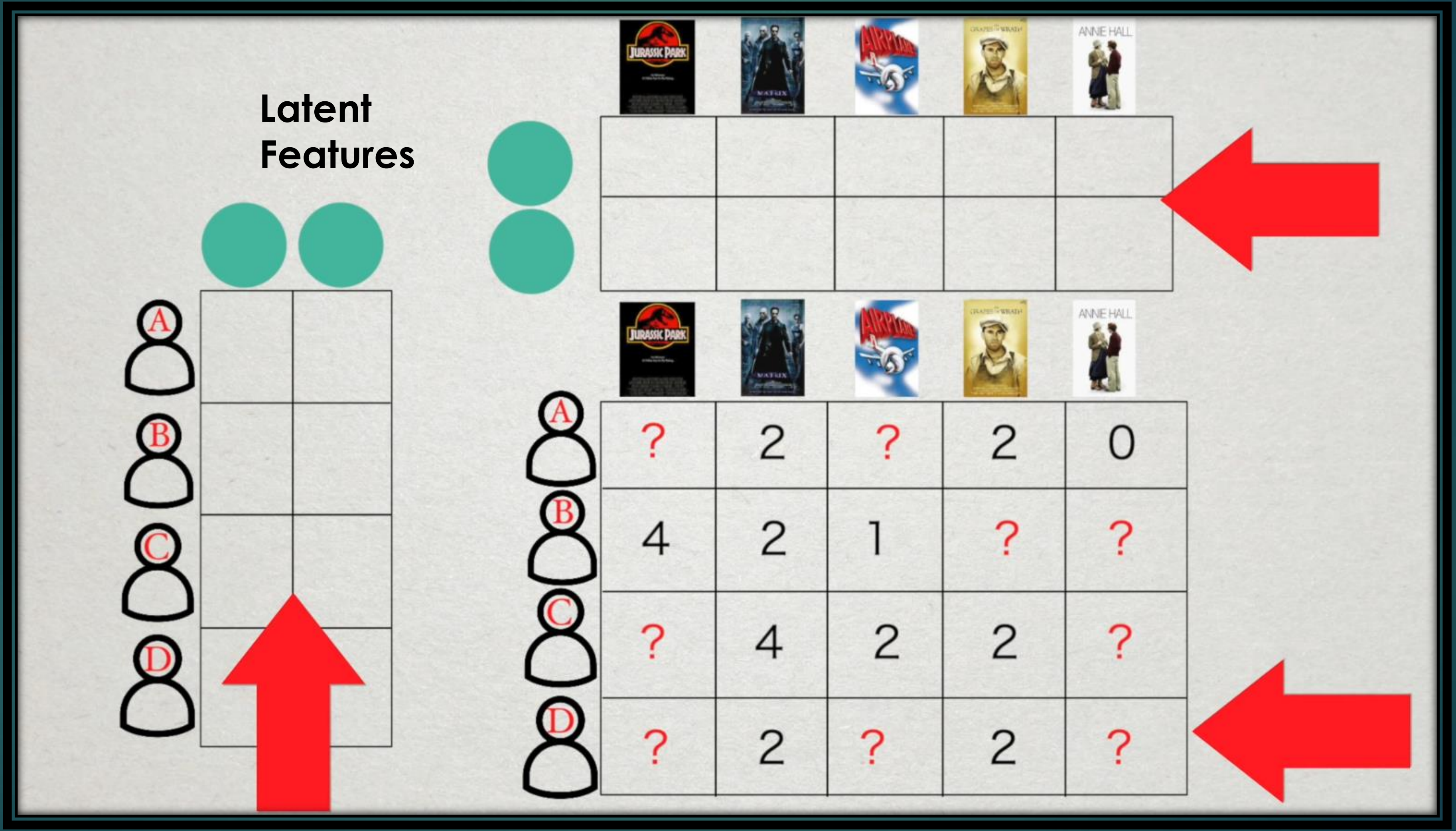
Collaborative filtering is an information retrieval method that recommends items to users based on how other users with **similar preferences** and behavior have interacted with that item.

"The users who liked this movie also liked that movie"

Collaborative Filtering - ALS



Collaborative Filtering - ALS



Collaborative Filtering - Implementation

- **Algorithm:** Alternating Least Squares (ALS)

- Python and PySpark for scalability
  - PySpark inherently **scalable**.
  - Scales to millions of rows with **cluster resources**.
  - Distributed Spark enables **linear scaling**.
  - **Outperforms** non-distributed frameworks.

- **Model Training**:

  - ALS implementation in PySpark MLlib

  - Hyperparameter tuning(rank, iterations)

Collaborative Filtering - Results

Root-mean-square error = **0.87**

===== User 414's Highly Rated Movies (Rating >= 4.0) =====

movieId	title	genres	rating
94	Beautiful Girls (1996)	Comedy|Drama|Romance	5.0
318	Shawshank Redemption, The (1994)	Crime|Drama	5.0
110	Braveheart (1995)	Action|Drama|War	5.0
223	Clerks (1994)	Comedy	5.0
296	Pulp Fiction (1994)	Comedy|Crime|Drama|Thriller	5.0
260	Star Wars: Episode IV - A New Hope (1977)	Action|Adventure|Sci-Fi	5.0
34	Babe (1995)	Children|Drama	5.0
266	Legends of the Fall (1994)	Drama|Romance|War|Western	5.0
11	American President, The (1995)	Comedy|Drama|Romance	5.0
290	Once Were Warriors (1994)	Crime|Drama	5.0

Precision@10 = **0.65**

===== Top 10 Recommended Movies for User 414 =====

movieId	title	genres	prediction
3379	On the Beach (1959)	Drama	5.1598325
96004	Dragon Ball Z: The History of Trunks (Doragon bôru Z: Zetsubô e no hankô!! Nokosareta chô senshi - Gohan to Torankusu) (1993)	Action|Adventure|Animation	5.1598325
33649	Saving Face (2004)	Comedy|Drama|Romance	4.9726944
102217	Bill Hicks: Revelations (1993)	Comedy	4.8807864
132333	Seve (2014)	Documentary|Drama	4.8510456
60943	Frozen River (2008)	Drama	4.832929
59018	Visitor, The (2007)	Drama|Romance	4.832929
6201	Lady Jane (1986)	Drama|Romance	4.807894
8235	Safety Last! (1923)	Action|Comedy|Romance	4.807894
171495	Cosmos	(no genres listed)	4.744145

Collaborative Filtering - Results

- **RMSE of 0.87** means, on average, our model's prediction deviates from the actual user rating by less than one star.

- the **Precision@10 of 0.65** means that, on average, 6.5 out of the top 10 movies recommended by the system were relevant items that the user had actually rated highly.

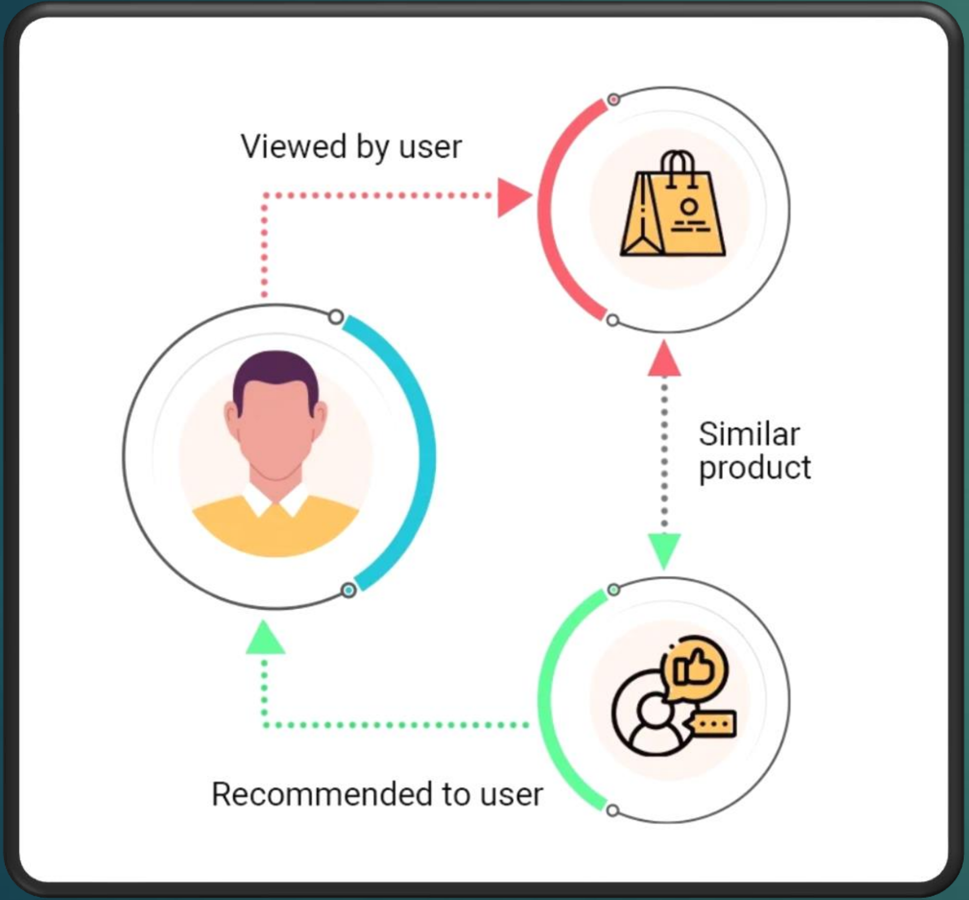
- While state-of-the-art systems on optimized datasets might achieve lower errors, an average error below one star is often considered practically useful in real-world scenarios.

- it's crucial to remember that the true effectiveness of any recommender system ultimately requires evaluation in real-world scenarios with actual users, as offline metrics like RMSE only capture part of the complex picture.

Collaborative Filtering – A Challenge!

- The "**cold start problem**" is a common challenge that occurs in recommender systems.
- It refers to **a situation where a system or algorithm runs into difficulties when it has little or no historical data about a user or an item**.
- Obviously, this makes it challenging to provide relevant personalized recommendations.

Content-Based Filtering



Recommends movies **similar** to what a user has liked in the past, based on movie content.

"If you liked this movie, you might like these similar movies..."

Content-Based Filtering - Feature Engineering

- **Genres** (binary features for all genres(19))

- **Decade** (binary features for decades[1990, 200, 2010])

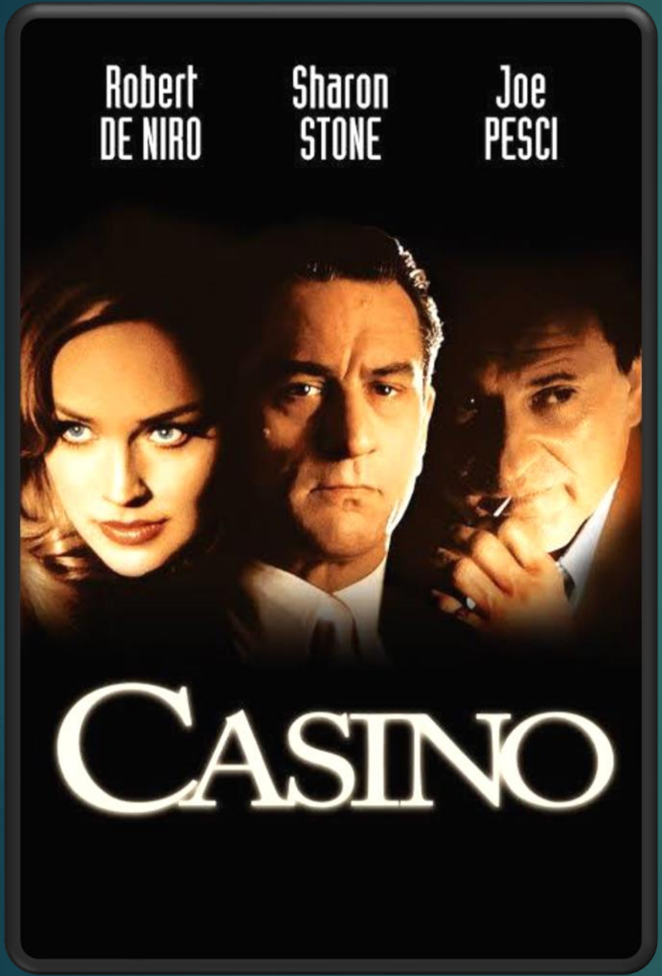
- **Tags** (binary features for top 200 tags)

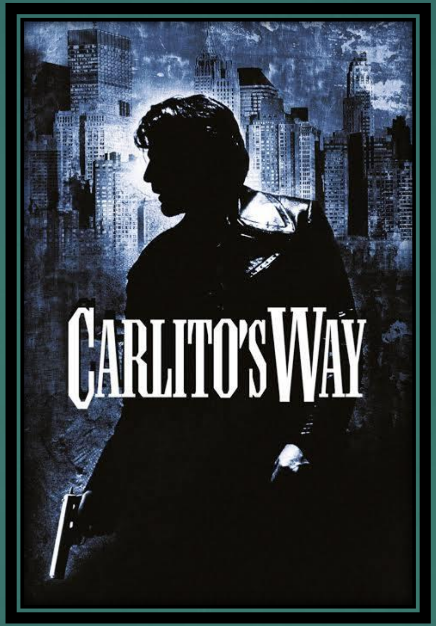
- Combine all of these features in one column called features which is a **vector**

- calculate **cosine similarity** between these "features" vectors to find the similar movies

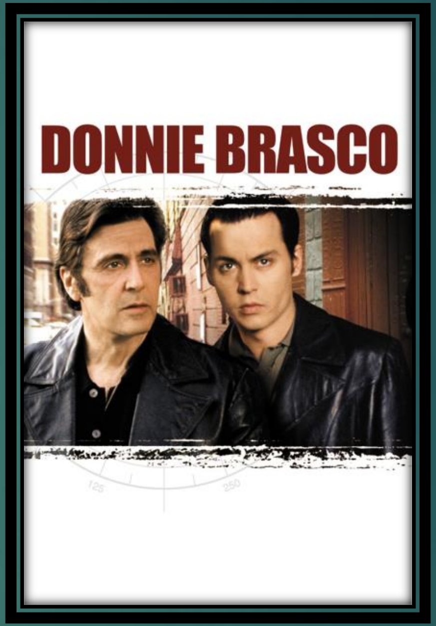
genre_Adventure	genre_Crime	genre_Sci-Fi	genre_Horror	genre_Fantasy
1	0	0	0	1
1	0	0	0	1
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	1	0	0	0
0	0	0	0	0
1	0	0	0	0
0	0	0	0	0
1	0	0	0	0
0	0	0	0	0
0	0	0	1	0
1	0	0	0	0
0	0	0	0	0
1	0	0	0	0
0	1	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	1	0	0	0

Content-Based Filtering - Example

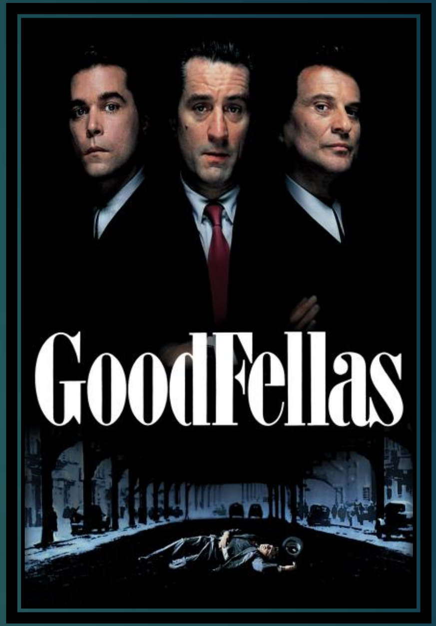




Similarity : 0.89



Similarity : 1



Similarity : 1

CF and CBF

Collaborative Filtering (CF):

- Strengths: Discover **unexpected** recommendations, works well with user interaction data.
- Weaknesses: **Cold start problem**.

Content-Based Filtering (CBF):

- Strengths: No cold start for new items, explainable recommendations based on content.
- Weaknesses: Relies on content quality, can be overly specific/less diverse recommendations.

Conclusion and Reflection

- Successfully built both **CF and CBF** movie recommendation systems using Spark.
- Demonstrated personalized user recommendations with CF and similar movie recommendations with CBF.
- Evaluated models using RMSE and Precision@k for CF.
- **Hybrid Recommender Systems**: Implement and evaluate hybrid approaches (weighted, switching, etc.) to combine CF and CBF strengths.
- **More Extensive Hyperparameter Tuning:** For CF.

References and Study Material

- Google Developers. (n.d.). Collaborative filtering basics. Retrieved from https://developers.google.com/machine-learning/recommendation/collaborative/basics

- GroupLens Research. (n.d.). MovieLens datasets. Retrieved from https://grouplens.org/datasets/movielens/

- Marketsy.ai. (2024, June 18). Hybrid recommender systems: Beginner's guide. Retrieved from https://marketsy.ai/blog/hybrid-recommender-systems-beginners-guide

- IBM. (2024, March 21). What is collaborative filtering? Retrieved from https://www.ibm.com/think/topics/collaborative-filtering

- ArtoftheProblem. (2020). [Recommender systems (Netflix/Amazon)] [Video]. YouTube. Retrieved from https://www.youtube.com/watch?v=n3RKsY2H-NE&ab_channel=ArtoftheProblem

- Serrano Academy. (2018). [How does Netflix recommend movies? Matrix factorization] [Video]. YouTube. Retrieved from https://www.youtube.com/watch?v=ZspR5PZemcs&t=483s&ab_channel=Serrano.Academy

- jamenlong1. (2017). [Recommendation engines using ALS in PySpark (MovieLens Dataset)] [Video]. YouTube. Retrieved from https://www.youtube.com/watch?v=FgGjc5oabrA&ab_channel=jamenlong1

- PyConCanada. (2019). [How to design and build a recommendation system pipeline in Python (Jill Cates)] [Video]. YouTube. Retrieved from https://www.youtube.com/watch?v=v_mONWiFv0k&ab_channel=PyConCanada

The End.