Movie Recommendation System Via SVD

Presentation by:
- Mário Silva nºmec 93430
- Daniel Gomes nºmec 93015
- Bruno Bastos nºmec 93302

Problem Description

With the abundance of data in recent years, interesting challenges are posed in the area of recommendation systems.

Producing high quality recommendations with scalability and performance is a major priority

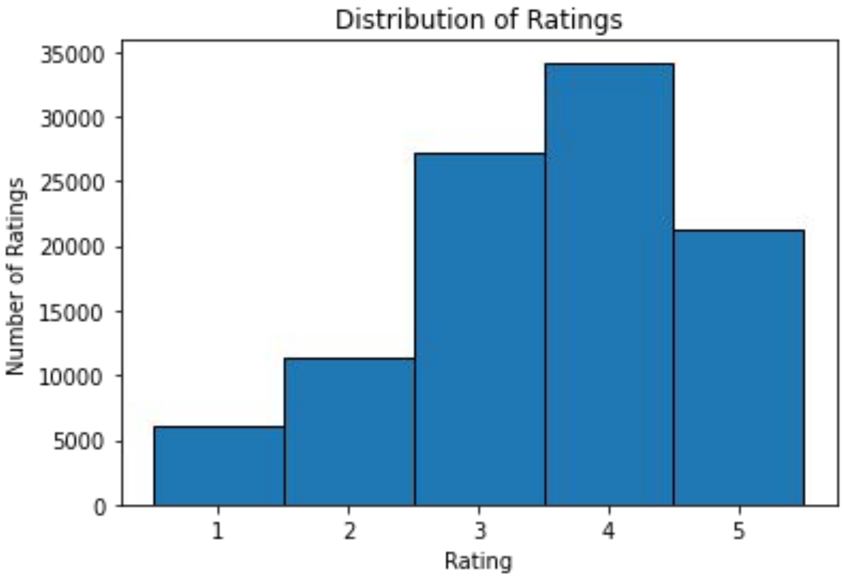
Singular Value Decomposition(SVD) based recommendation algorithms have been leveraged to produce better results.

Dataset

Movie Lens Dataset:

- 100 000 movie ratings in a 1-5 scale
- At least, 20 ratings per user
- 943 users and 1682 different movies
- Data separated in a ratio of 80%/20% for the training and test data, respectively
    - For the test data, random user-movie ratings are retrieved from the data

Dataset Analysis



	rating
count	100000.000000
mean	3.529860
std	1.125674
min	1.000000
25%	3.000000
50%	4.000000
75%	4.000000
max	5.000000

Dataset Analysis

Sparsity of the Ratings matrix: **93.7**%

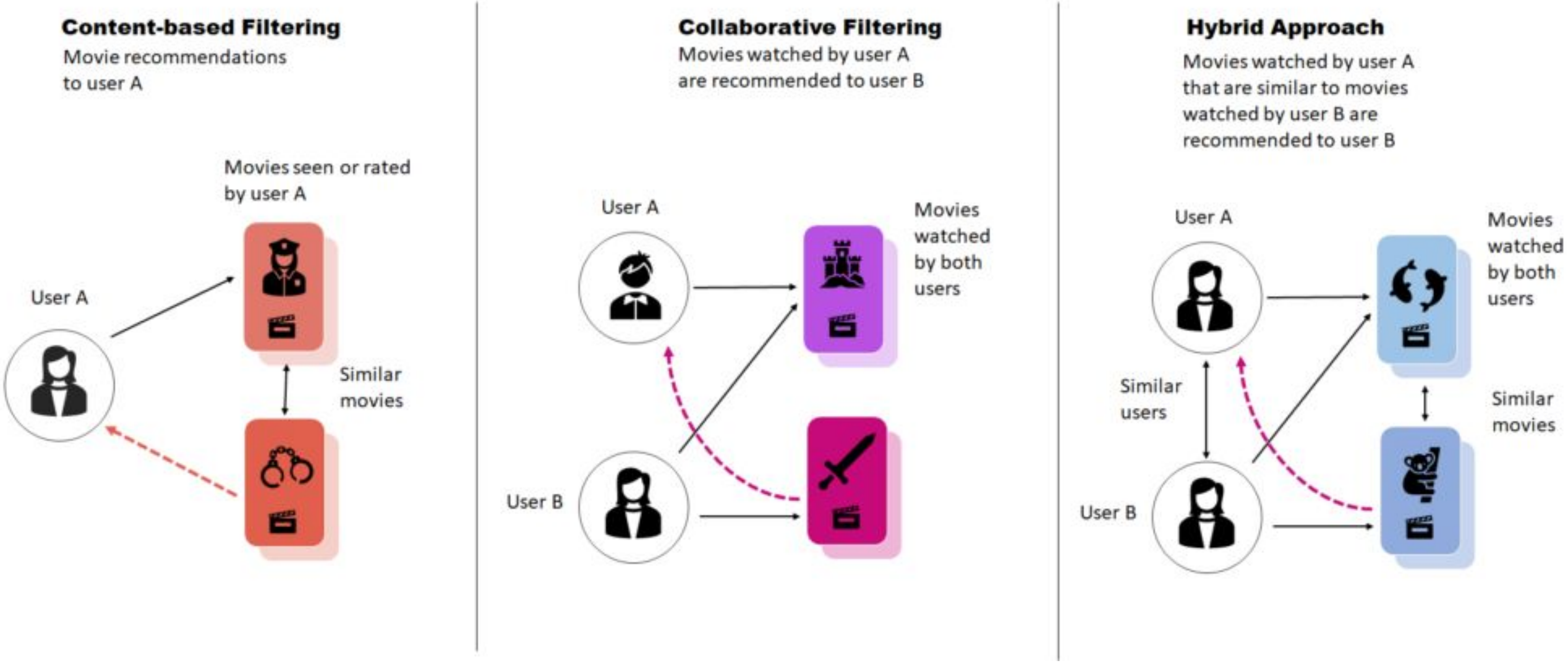
Movies IDs

Users IDs

	0	1	2	3	4
0	5.0	3.0	4.0	3.0	3.0
1	4.0	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN
4	4.0	3.0	NaN	NaN	NaN

Sample of the Ratings Matrix

Recommendation System



Memory vs Model Based Techniques

Memory Based

Find similar users/items based on similarity measures and take weighted average of ratings to recommend unrated items

Advantage
Easy creation and explainability of results

Disadvantage
Performance reduces when data is sparse. Non-scalable.

VS

Model Based

Use machine learning to find user ratings of unrated items

Advantage
Dimensionality reduction deals with missing data

Disadvantage
Inference is intractable because of hidden/latent factors

Collaborative Filtering - Pros & Cons

Pros

Captures inherent subtle characteristics

Suggest unique items by observing similar-minded peoples behavior.

Make a real quality assessment of items by considering people's experience

Cons

Sparsity of the rating matrix

Cold-start

Spam Attacks

SVD - Singular Value Decomposition

$$\underset{m \times n}{A} = \underset{m \times m}{U} \; \underset{m \times m}{S} \; \underset{m \times n}{V^T}$$

A: Data Matrix

- **m users**, **n movies**

In our case:

m < n

U: Left Singular Vectors

- **Users x Singular Values**

S: Singular Values

- Diagonal matrix with **strength of each singular value**

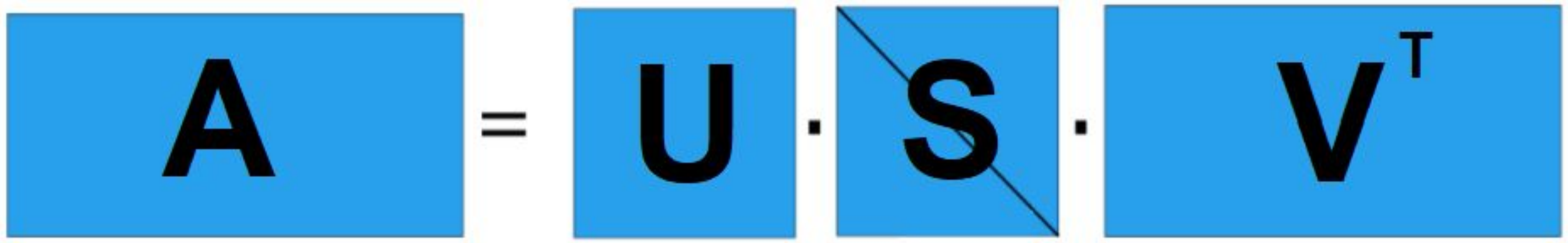
V: Right Singular Vectors

- **Movies x Singular Values**

SVD - Singular Value Decomposition

It is always possible to decompose a real matrix with SVD where:

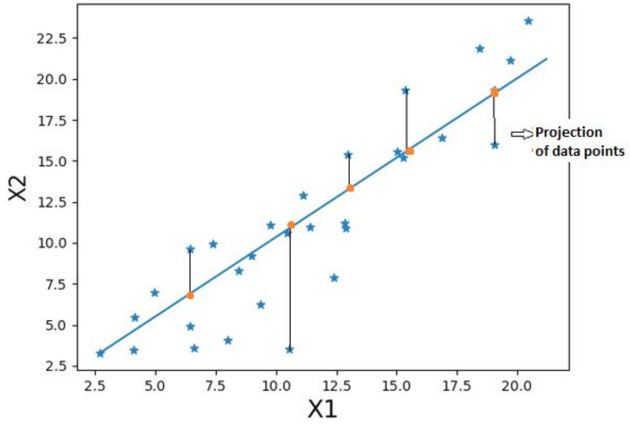
- U, S, V: **unique**
- U, V: **column orthonormal**
- S: diagonal
    - Entries (singular values) are positive and sorted in decreasing order

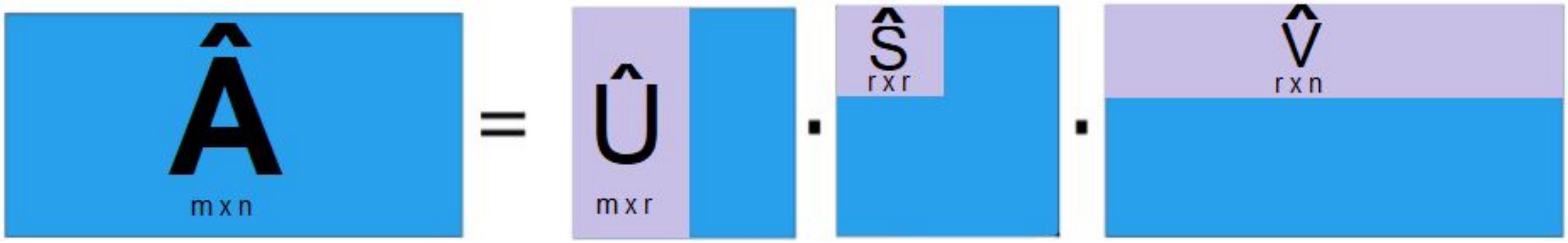


m x n | m x m | m x m | m x n

Dimensionality Reduction

- SVD finds linear correlations
- Removing singular values from the matrix S reduces the dimensionality of the data
- The resulting data matrix will be an approximation of the initial





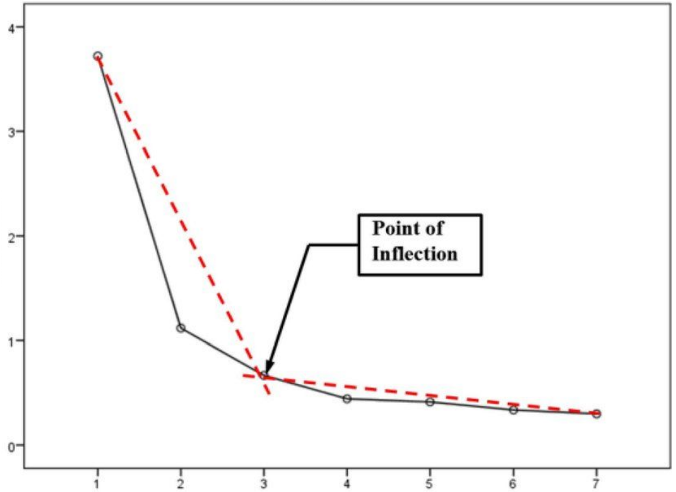
Truncated SVD

- Improves performance
- Keeps the latent factors that best represent the data
- Reduces overfitting of the model

However, this comes with an **increased reconstruction error** of the original matrix.

How to find the optimal number of latent factors?

- Keep 80-90 % of 'energy'
- Elbow Method



Query with SVD

SVD needs to consider unrated items, thus its value can be initialized using the following approaches:

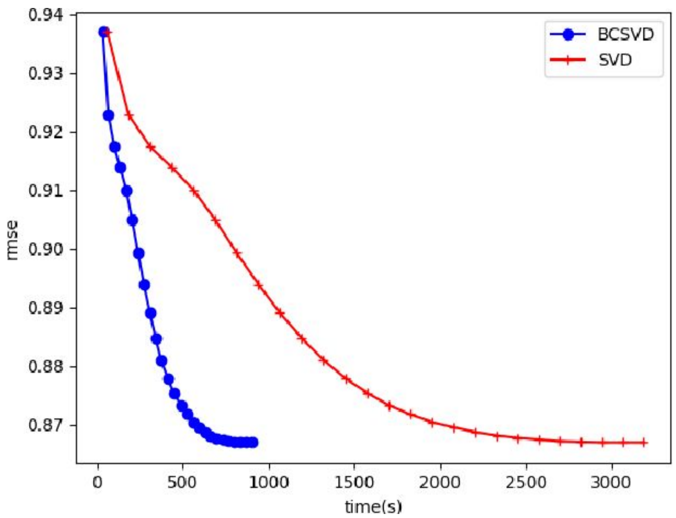
- Use zeros to identify each unrated item
- Global Mean
- Local Mean

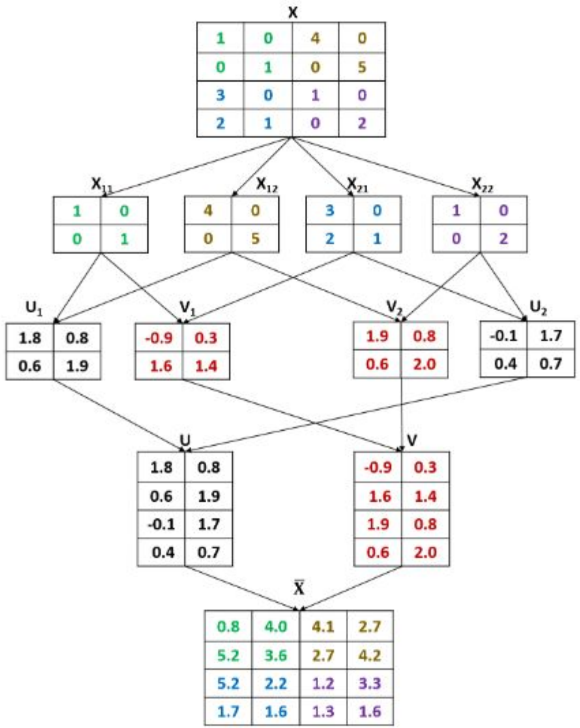
However, these methods do not produce good results when the matrix is very sparse.

Block Based SVD

Sometimes calculating SVD can be computationally expensive and can take a long time.

Block Based SVD proposes a distributed approach by splitting the matrix in blocks and calculating each block using concurrency.





Latent Factor Recommendation System

SVD isn't defined when entries are missing!
Use specialized methods to find P, Q:

$$\min_{P,Q} \sum_{(i,x)\in R} \left(r_{xi} - q_i \cdot p_x\right)^2$$

Example: **Funk SVD**

1		3			5			5		4	
		5	4			4			2	1	3
2	4		1	2		3		4	3	5	
	2	4		5			4			2	
		4	3	4	2					2	5
1		3		3			2			4	

≈

factors

.1	-.4	.2
-.5	.6	.5
-.2	.3	.5
1.1	2.1	.3
-.7	2.1	-2
-1	.7	.3

Q

1.1	-.2	.3	.5	-2	-.5	.8	-.4	.3	1.4	2.4	-.9
-.8	.7	.5	1.4	.3	-1	1.4	2.9	-.7	1.2	-.1	1.3
2.1	-.4	.6	1.7	2.4	.9	-.3	.4	.8	.7	-.6	.1

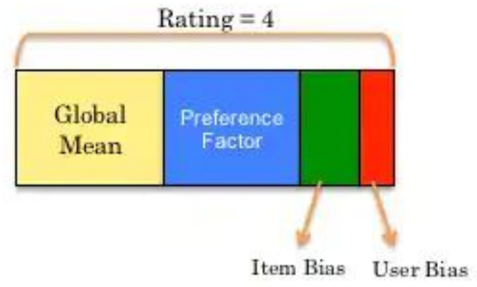
factors

P^T

Stochastic Gradient Descent (SGD)

Goals:
- Minimize errors
- Prevent overfitting

$$r_{xi} = \underset{\substack{\text{Overall}\\\text{mean rating}}}{\mu} + \underset{\substack{\text{Bias for}\\\text{user } x}}{b_x} + \underset{\substack{\text{Bias for}\\\text{movie } i}}{b_i} + \underset{\substack{\text{User-Movie}\\\text{interaction}}}{q_i \cdot p_x}$$



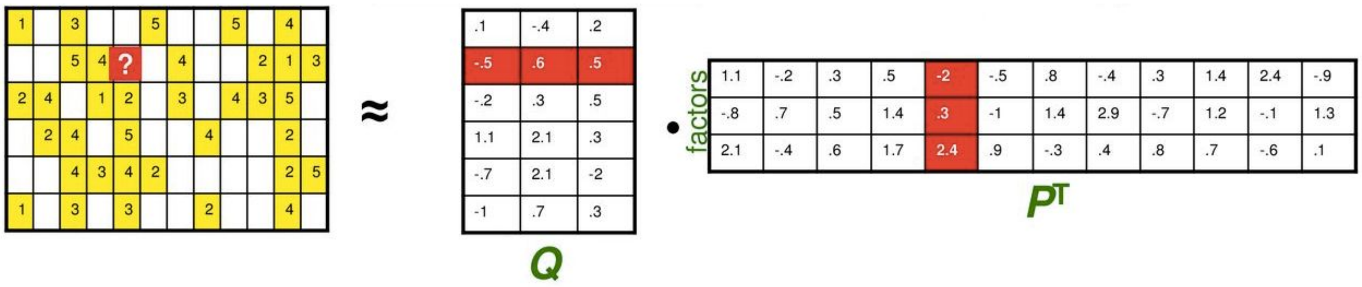
- **SGD** can be used to find parameters by solving:

$$\min_{Q,P} \underbrace{\sum_{(x,i)\in R} \left(r_{xi} - (\mu + b_x + b_i + q_i\, p_x)\right)^2}_{\text{goodness of fit}} + \underbrace{\left(\lambda_1 \sum_i \|q_i\|^2 + \lambda_2 \sum_x \|p_x\|^2 + \lambda_3 \sum_x \|b_x\|^2 + \lambda_4 \sum_i \|b_i\|^2\right)}_{\text{regularization}}$$

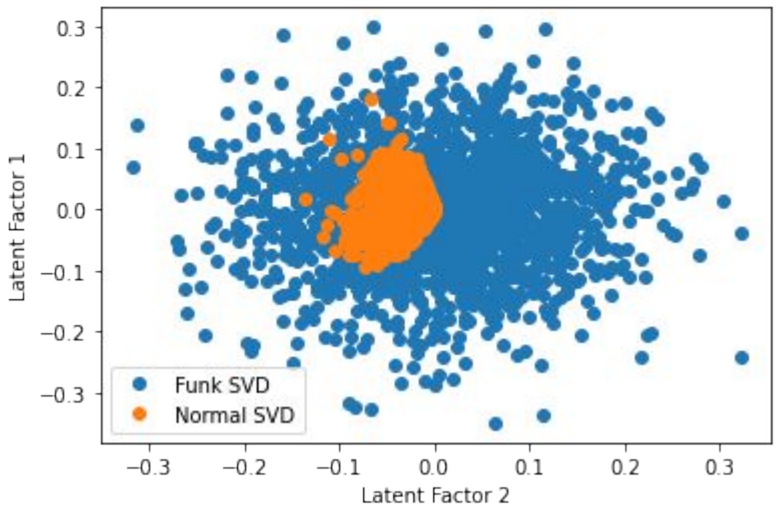
λ is selected via grid-search on a validation set

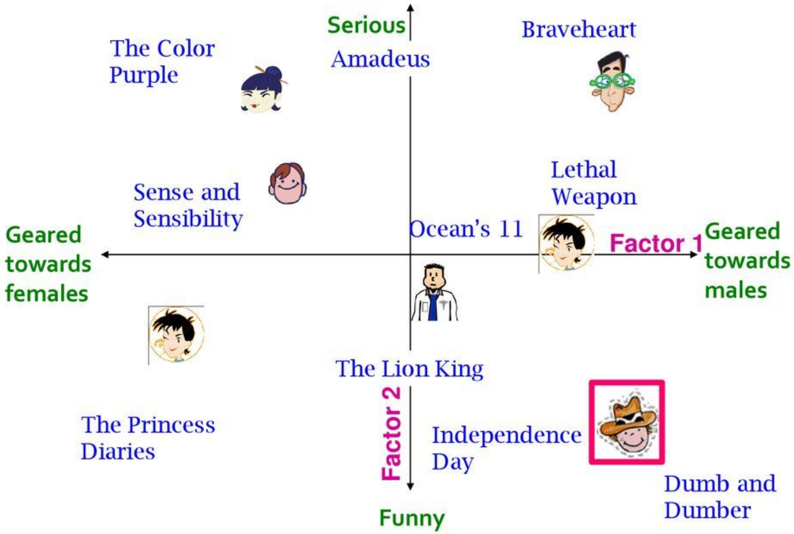
Query with Funk SVD

$$r_{xi} = \underbrace{\mu}_{\text{Overall mean rating}} + \underbrace{b_x}_{\text{Bias for user } \boldsymbol{x}} + \underbrace{b_i}_{\text{Bias for movie } \boldsymbol{i}} + \underbrace{q_i \cdot p_x}_{\text{User-Movie interaction}}$$

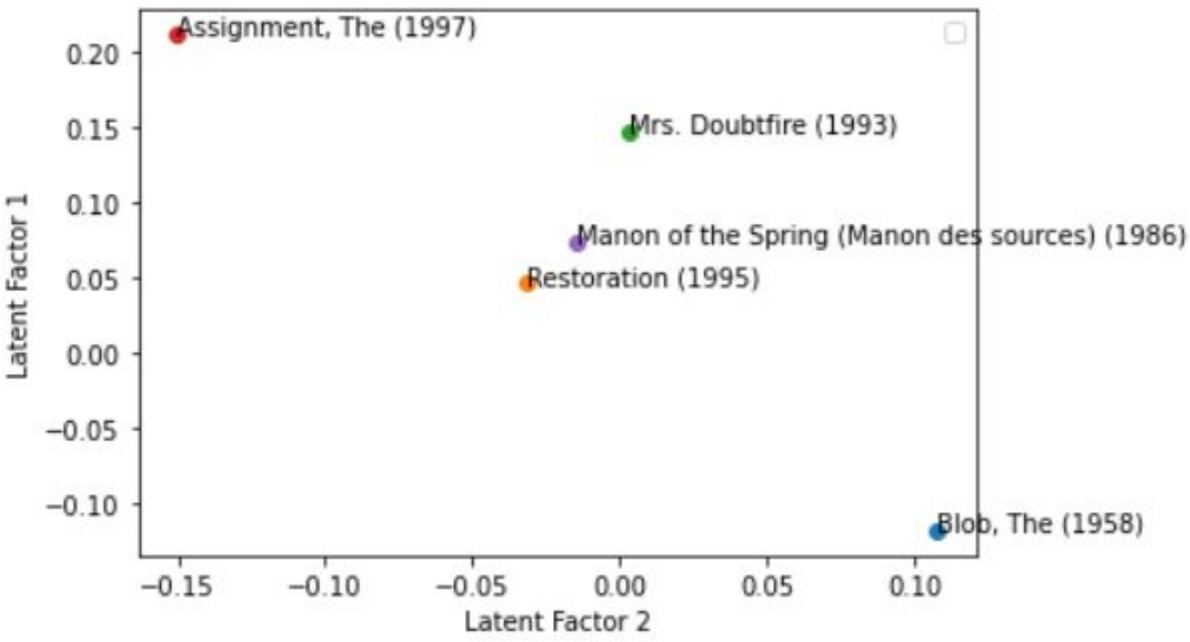


Similarity Analysis





Clustering Possibility



References

1. https://arxiv.org/pdf/1907.07410.pdf
2. https://www.researchgate.net/publication/346651834_Various_Types_of_Movies_Recommendation_System
3. https://www.relataly.com/building-a-movie-recommender-using-collaborative-filtering/4376/
4. https://machinelearningmastery.com/using-singular-value-decomposition-to-build-a-recommender-system/
5. https://analyticsindiamag.com/singular-value-decomposition-svd-application-recommender-system/
6. https://towardsdatascience.com/recommender-system-singular-value-decomposition-svd-truncated-svd-97096338f361
7. https://towardsdatascience.com/beginners-guide-to-creating-an-svd-recommender-system-1fd7326d1f65
8. https://theory.stanford.edu/~tim/s15/l/l9.pdf
9. https://medium.datadriveninvestor.com/how-funk-singular-value-decomposition-algorithm-work-in-recommendation-engines-36f2fbf62cac