Università degli studi di Trieste

Data Science and Scientific Computing

Movie Recommendation

Recommendation System for MovieLens Dataset

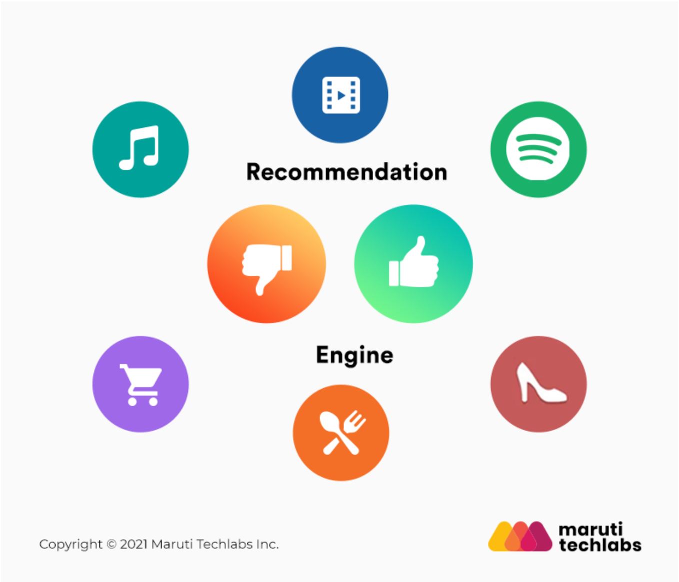
Final Project – Information Retrieval Course

Babaei Elham

Recommendation Systems

- **Recommendation system (Recommender system):**
  - A subset of information filtering system that seeks predict the “rating” or ”preference” a user would give to an item.

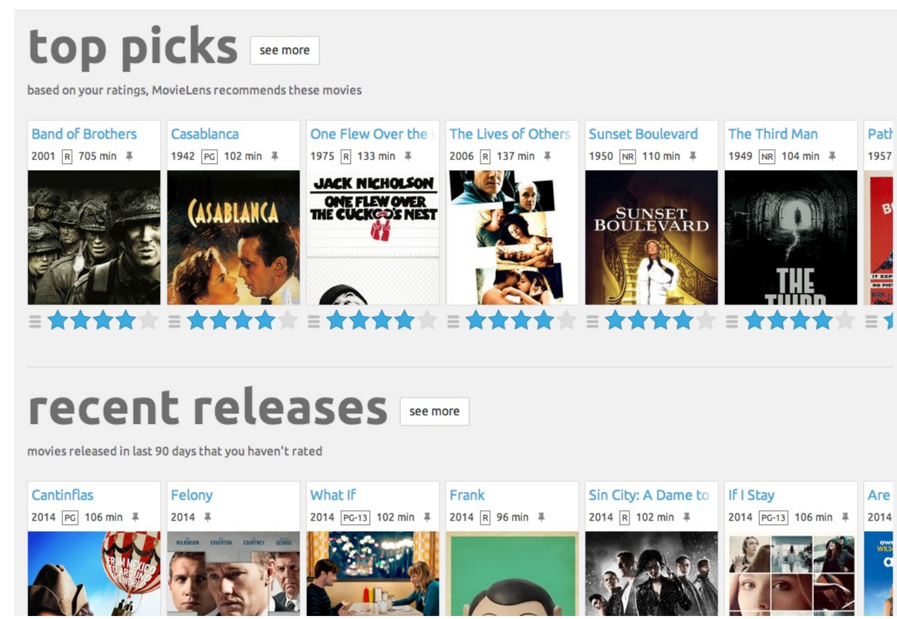
- **Applications**
  - Entertainment industry such as music and movie
  - Books and articles
  - Websites
  - Tourism spots
  - Restaurants
  - Online dating etc.



Dataset

- MovieLens dataset is collected from https://movielens.org/ by https://grouplens.org/

- 9742 movies
- 610 users
- Star rating (0.5 - 5)
- Includes four csv files Movies, Ratings, Tags, and Links.



Dataset

```
(9742, 3)
```

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

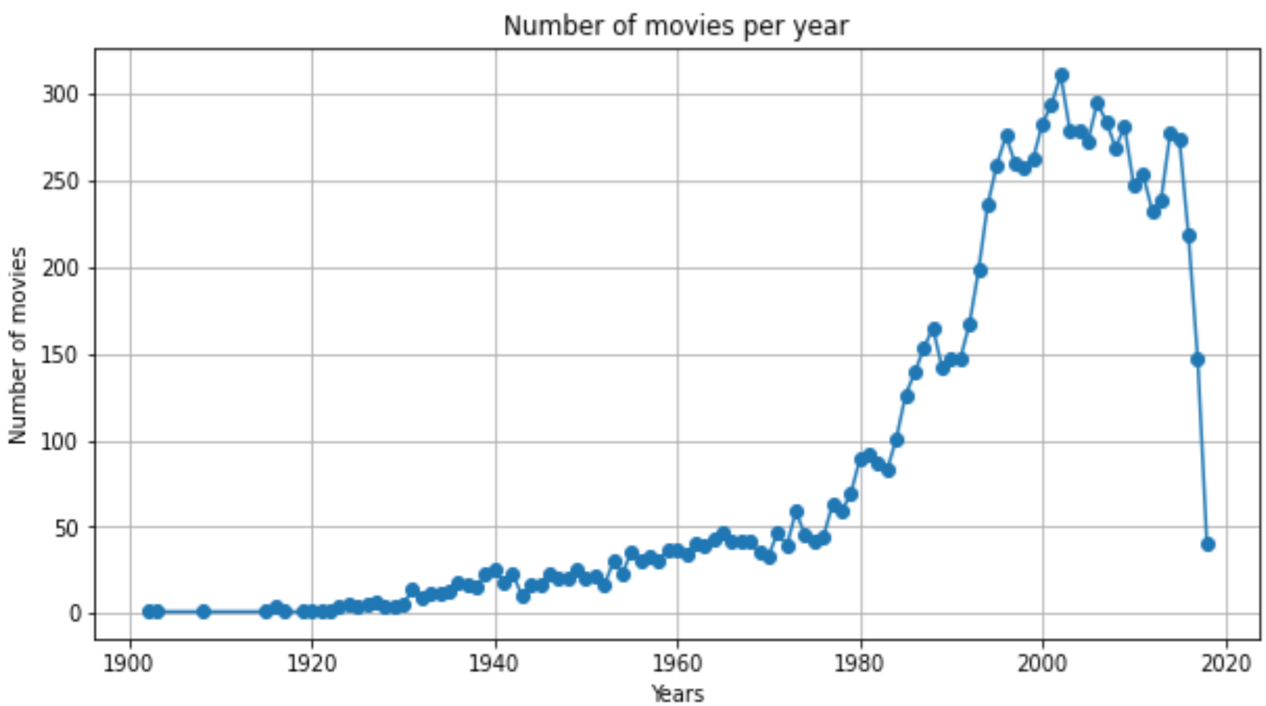
```
(100836, 3)
```

	userId	movieId	rating
0	1	1	4.0
1	1	3	4.0
2	1	6	4.0
3	1	47	5.0
4	1	50	5.0

```
(3683, 3)
```

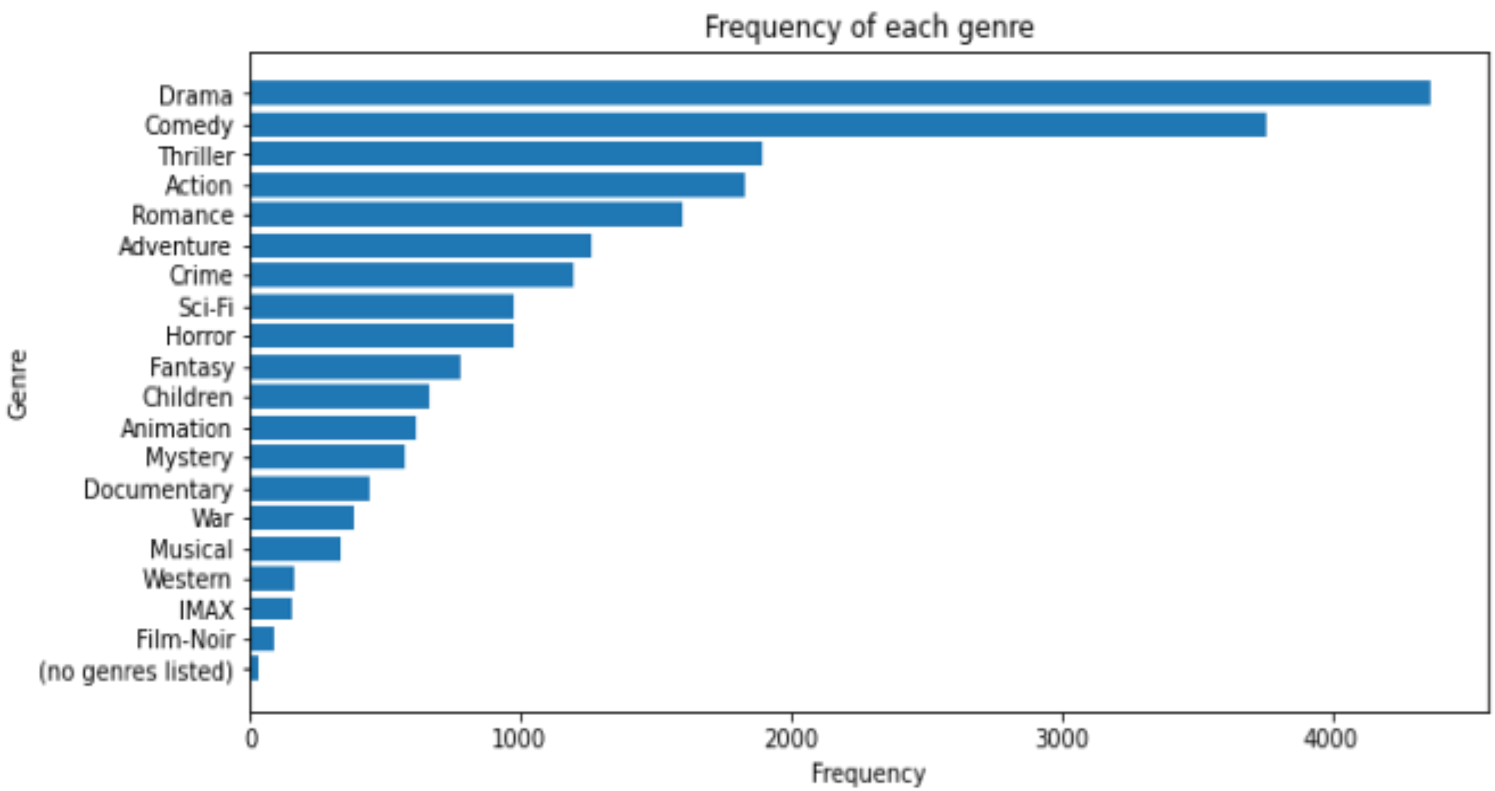
	userId	movieId	tag
0	2	60756	funny
1	2	60756	Highly quotable
2	2	60756	will ferrell
3	2	89774	Boxing story
4	2	89774	MMA

Dataset



Max No. of Movies Released = 311
Year = 2002

Dataset



Frequency of each genre
Drama
Comedy
Thriller
Action
Romance
Adventure
Crime
Sci-Fi
Horror
Fantasy
Children
Animation
Mystery
Documentary
War
Musical
Western
IMAX
Film-Noir
(no genres listed)
Genre
0
1000
2000
3000
4000
Frequency

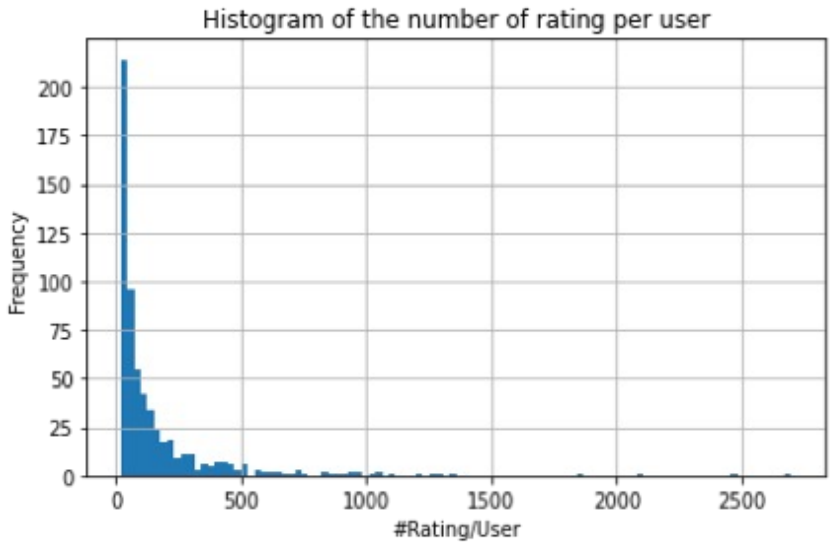
Ratings/Feedback Matrix

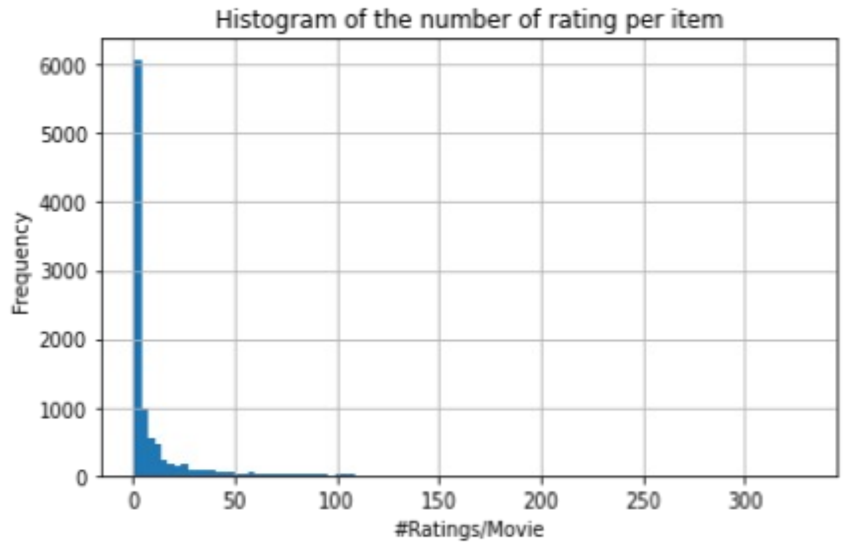
- Ratings matrix is a matrix in which the rating given by each user to each movie is included.
- We need this matrix to build our recommender system. In order to create it, we first merge the two datasets Movies and Ratings:

	userId	movieId	rating
count	100836.000000	100836.000000	100836.000000
mean	326.127564	19435.295718	3.501557
std	182.618491	35530.987199	1.042529
min	1.000000	1.000000	0.500000
25%	177.000000	1199.000000	3.000000
50%	325.000000	2991.000000	3.500000
75%	477.000000	8122.000000	4.000000
max	610.000000	193609.000000	5.000000

- The size of the result ratings matrix is 610*9724
- Sparsity percentage of ratings matrix: 98.3% !

Dataset

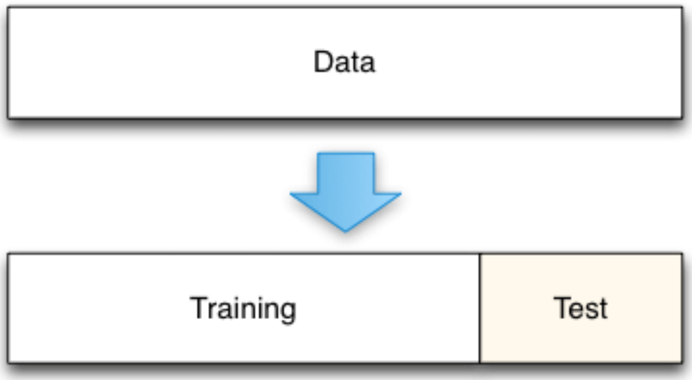

Histogram of the number of rating per user
Frequency
200
175
150
125
100
75
50
25
0
0
500
1000
1500
2000
2500
#Rating/User


Histogram of the number of rating per item
Frequency
6000
5000
4000
3000
2000
1000
0
0
50
100
150
200
250
300
#Ratings/Movie

Train/Test split

- To split data into the train and test datasets, we just remove some available ratings(nonzero) from our ratings matrix and consider it as the test dataset

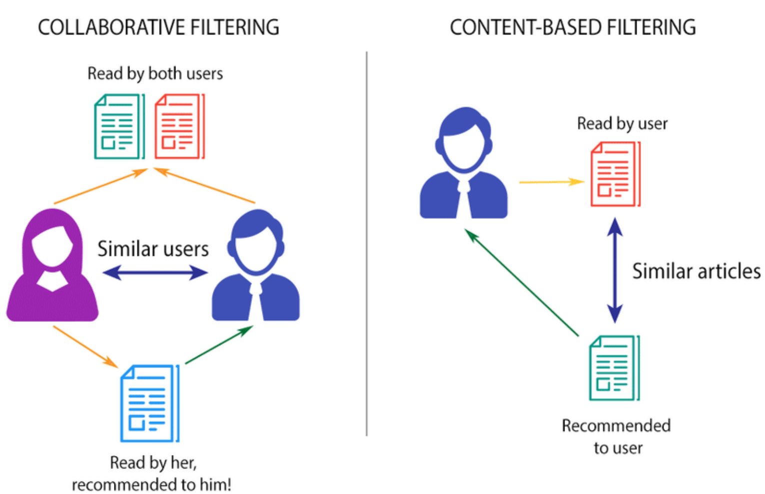
- Given we already know each user has given more than 10 ratings, what we do is for every user, we remove 10 of the movie ratings and assign them to the test dataset. As a result, the size of test and train datasets will be 610*9724.



Collaborative and Content Based Filtering

- Collaborative filtering is a method of making automatic predictions (**filtering**) about the interests of a user by collecting preferences or taste information from many users (**collaborating).**

- Content-based filtering uses similarity between item to recommend them to the user.



Collaborative and Content Based Filtering

Content Based Filtering

Advantages

- The model doesn't need any data about other users, since the recommendations are specific to this user. This makes it easier to scale to a large number of users.
- The model can capture the specific interests of a user, and can recommend niche items that very few other users are interested in.

Disadvantages

- Since the feature representation of the items are hand-engineered to some extent, this technique requires a lot of domain knowledge. Therefore, the model can only be as good as the hand-engineered features.
- The model can only make recommendations based on existing interests of the user. In other words, the model has limited ability to expand on the users' existing interests.

Collaborative Filtering

Advantages

- It doesn't need domain knowledge because the embeddings are automatically learned.
- The model can help users discover new interests. The system may not know the user is interested in a given item, but the model might still recommend it because similar users are interested in that item
- The system needs only the feedback matrix to train a matrix factorization model. In particular, the system doesn't need contextual features. (great starting point).

Disadvantages

- The prediction of the model for a given (user, item) pair is the dot product of the corresponding embeddings. So, if an item is not seen during training, the system can't create an embedding for it and can't query the model with this item (**cold-start problem**).
- Hard to include side features for query/item; e.g. age or country.

Embeddings

- Decomposing the ratings matrix into the product of two lower dimensionality rectangular matrices.

Ratings matrix : $C \subset R^{m \times n}$

User embeding: $U \subset R^{m \times k}$ → UV^T An approximation of the ratings matrix C

Item embedings: $V \subset R^{n \times k}$

K is the number **of latent factors** for which we do not know:

- The number of them
- What exactly they are

	M1	M2	M3	M4	M5
F1	1.2	3.1	0.3	2.5	0.2
F2	2.4	1.5	4.4	0.4	1.1

$1.2 \times 0.2 + 2.4 \times 0.5 = 1.44$

	F1	F2
A	0.2	0.5
B	0.3	0.4
C	0.7	0.8
D	0.4	0.5

	M1	M2	M3	M4	M5
	1.44	1.37	2.26	0.7	0.59
	1.32	1.53	1.85	0.91	0.5
	2.76	3.37	3.73	2.07	1.02
	1.68	1.99	2.32	1.2	0.63

↔

	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

Embeddings

- **How to obtain the embeddings U and V ?**
  - By minimizing one of the below objective functions:
    - Singular Value Decomposition (SVD)

$$\min_{U \subset R^{m\times k}, V \subset R^{n\times k}} \quad L = |C - UV^T|_F^2 = \sum_{(i,j)} \left(C_{ij} - U_i . V_j\right)^2$$

    - Observed-only matrix factorization

$$\min_{U \subset R^{m\times k}, V \subset R^{n\times k}} \quad L = \sum_{(i,j)\in obs} \left(C_{ij} - U_i . V_j\right)^2$$

    - Weighted matrix factorization (weighted MF)

$$\min_{U \subset R^{m\times k}, V \subset R^{n\times k}} \quad L = \sum_{(i,j)\in obs} \left(C_{ij} - U_i V_j\right)^2 + W_0 \sum_{(i,j)\notin obs} \left(0 - U_i . V_j\right)^2$$

Weighted MF and WALS

$$\min_{U \subset R^{m\times k}, V \subset R^{n\times k}} \quad L = \sum_{(i,j)\in obs} \left(C_{ij} - U_i V_j\right)^2 + W_0 \sum_{(i,j)\notin obs} \left(0 - U_i . V_j\right)^2$$

- **How to minimize the loss function ?**
  - Stochastic Gradient Descent (SGD)
  - Weighted Alternating Least Squares (WALS)
    - Start with U and V randomly generated.
    - Fix U and find V.
    - Fix V and find U.
    - Repeat until convergence.

$$\frac{\partial L}{\partial V_j} = 0 \implies V_j = C_j U (U^T U + W_0 I)^{-1}$$

$$\frac{\partial L}{\partial U_i} = 0 \implies U_i = C_i V (V^T V + W_0 I)^{-1}$$

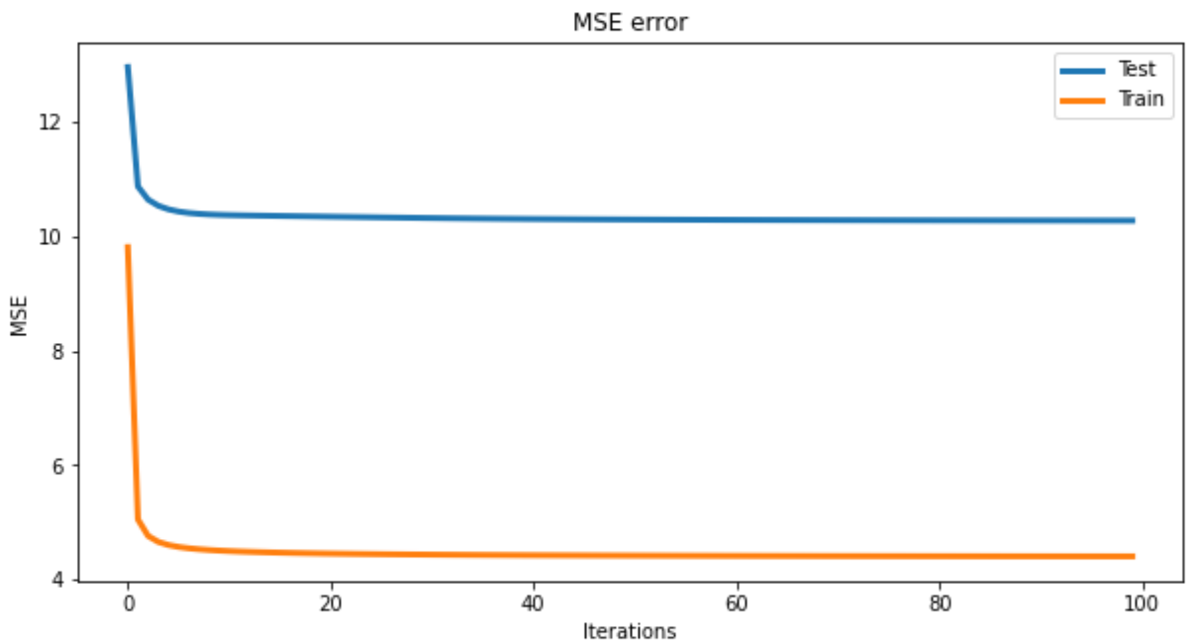
Building The Model

- **Assumptions:**
  - Random Values are generated from a standard normal distribution to initialize the matrices U and V in WALS algorithm.
  - W_0=0.1
  - The number of latent factors = 40
  - The number of Iterations = 100
  - Mean Square Error (MSE) to measure the performance.

Approximated C (ratings matrix)

```
U shape:  (610, 40)
V shape: (9724, 40)
C_approximated shape:  (610, 9724)
C_approximated:
          1         2         3    ...  193585    193587    193609
1    3.497251  0.381825  1.353872  ...     0.0 -0.026071 -0.060732
2    0.240155 -0.002077 -0.041792  ...     0.0  0.008247  0.017447
3   -0.010514  0.002361  0.057208  ...     0.0 -0.000798 -0.001453
4    1.843769 -0.085551 -0.239604  ...     0.0 -0.005918 -0.006021
5    1.064515  0.841579  0.309624  ...     0.0 -0.001353  0.003277
..        ...       ...       ...  ...     ...       ...       ...
606  2.692108  0.023498  0.181179  ...     0.0 -0.039771 -0.057492
607  2.865503  1.563101  0.538289  ...     0.0 -0.011397 -0.040497
608  1.038914  2.422161  3.107254  ...     0.0 -0.014384  0.036209
609  0.559981  0.525596  0.137997  ...     0.0  0.001760  0.004321
610  5.110386  0.021788 -0.094111  ...     0.0 -0.015881  0.048842
```

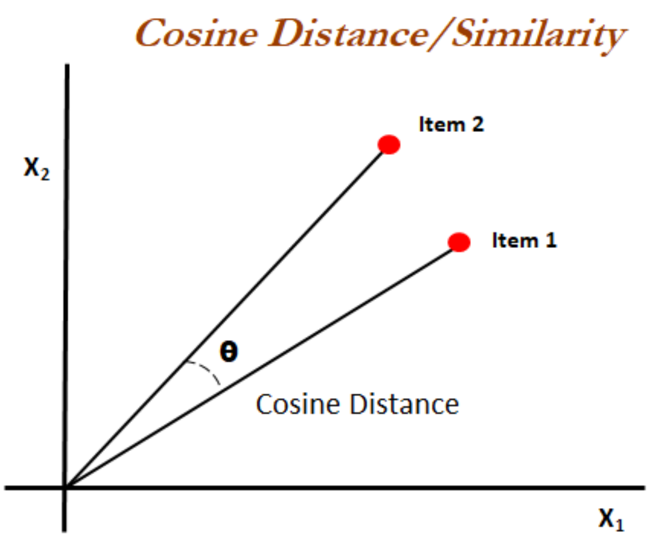
MSE Error



WALS converges very quickly!
Time consumed = 23 seconds

Similarity Metric

- Each column (movie) or row (user) in the approximated matrix C can be considered as a vector to compute the similarity between pairs of vectors; which are called **item similarity** and **user similarity** respectively.
- Similarity metrics:
  - Cosine similarity
  - Dot product
  - Euclidean distance



$$\text{cosine similarity} = S_C(A,B) := \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}},$$

Similarity Matrix

movieId	1	2	3	4	5	6	7	8
movieId								
1	1.000000	0.722542	0.498184	0.288227	0.555568	0.590114	0.480009	0.297563
2	0.722542	1.000000	0.602541	0.466215	0.661088	0.474644	0.626206	0.430237
3	0.498184	0.602541	1.000000	0.514972	0.737405	0.470003	0.678786	0.464228
4	0.288227	0.466215	0.514972	1.000000	0.687901	0.345505	0.757356	0.411968
5	0.555568	0.661088	0.737405	0.687901	1.000000	0.425396	0.752628	0.434034
...	...	...	...	...	...	...	...	...
193581	0.027856	0.108683	-0.020702	0.084082	0.114082	-0.064087	0.085955	-0.028131
193583	0.027856	0.108683	-0.020702	0.084082	0.114082	-0.064087	0.085955	-0.028131
193585	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
193587	0.027856	0.108683	-0.020702	0.084082	0.114082	-0.064087	0.085955	-0.028131
193609	0.149908	0.035206	-0.035121	-0.111401	0.014838	0.084229	-0.096083	-0.021937

9724 rows × 9724 columns

Input Queries and Movies' Score

- If we have a new user who has given the following ratings to the movies with IDs 1, 3, 100, and 23:

```
new_user = [(1,5),(3,5),(100,2.5),(23,1)]
```

- We a similarity matrix that shows the similarity between each movie and the ones rated by the user:

	1	2	3	4	5	6	7	8	9	10	11	12
0	2.500000	1.806356	1.245460	0.720567	1.388921	1.475285	1.200024	0.743906	0.821041	1.601781	1.362800	0.956749
1	1.245460	1.506352	2.500000	1.287429	1.843511	1.175008	1.696964	1.160570	1.178835	1.337205	1.413411	0.887562
2	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
3	-0.466903	-0.642556	-0.488331	-0.429851	-0.624010	-0.768985	-0.535884	-0.980385	-0.265131	-0.844922	-0.574875	-0.497705

4 rows × 9724 columns

- The summation of each column in this matrix is the score of the associated movie.

Movies' Ranking

```
1       3.278556
3       3.257129
2054    3.191534
788     3.045911
586     3.034914
1073    2.979349
780     2.887071
2987    2.881919
39      2.879251
2804    2.872025
2406    2.871923
500     2.869991
2797    2.864688
480     2.860697
3421    2.853950
2174    2.845799
104     2.838188
736     2.826242
588     2.813096
2918    2.806585
dtype: float64
```

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
22	23	Assassins (1995)	Action|Crime|Thriller
88	100	City Hall (1996)	Drama|Thriller
622	788	Nutty Professor, The (1996)	Comedy|Fantasy|Romance|Sci-Fi
1522	2054	Honey, I Shrunk the Kids (1989)	Adventure|Children|Comedy|Fantasy|Sci-Fi

Further Studies

- Considering weights to the observed values in weighted MF.
- Using some theoretical approaches to tune the weights and the number of latent factors.
- Using neural networks to build the recommender system .
- Hybrid approaches (a combination of content-based and collaborative filtering).
- Dealing with recommender system as a classification problem.

References

- An introduction to information retrieval, Christopher D. Manning Prabhakar Raghavan Hinrich Schütze, 2009.
- https://developers.google.com
- *https://www.researchgate.net/figure/Train-Test-Data-Split_fig6_325870973*
- *https://www.researchgate.net/figure/Content-based-filtering-vs-Collaborative-filtering-Source_fig5_323726564*
- *Wikipedia.com*
- *https://www.oreilly.com/*
- https://www.youtube.com/channel/UCgBncpylJ1kiVaPyP-PZauQ

Thank you