



# Recommendation using Collaborative Filtering

By  
Chris Alexander  
Viveka Salinamakki

# What is a Recommendation System?

---



Predicts/Recommends the products or services that users would buy or consume



Used in e-commerce, entertainment content, social media posts, advertisements, music, etc

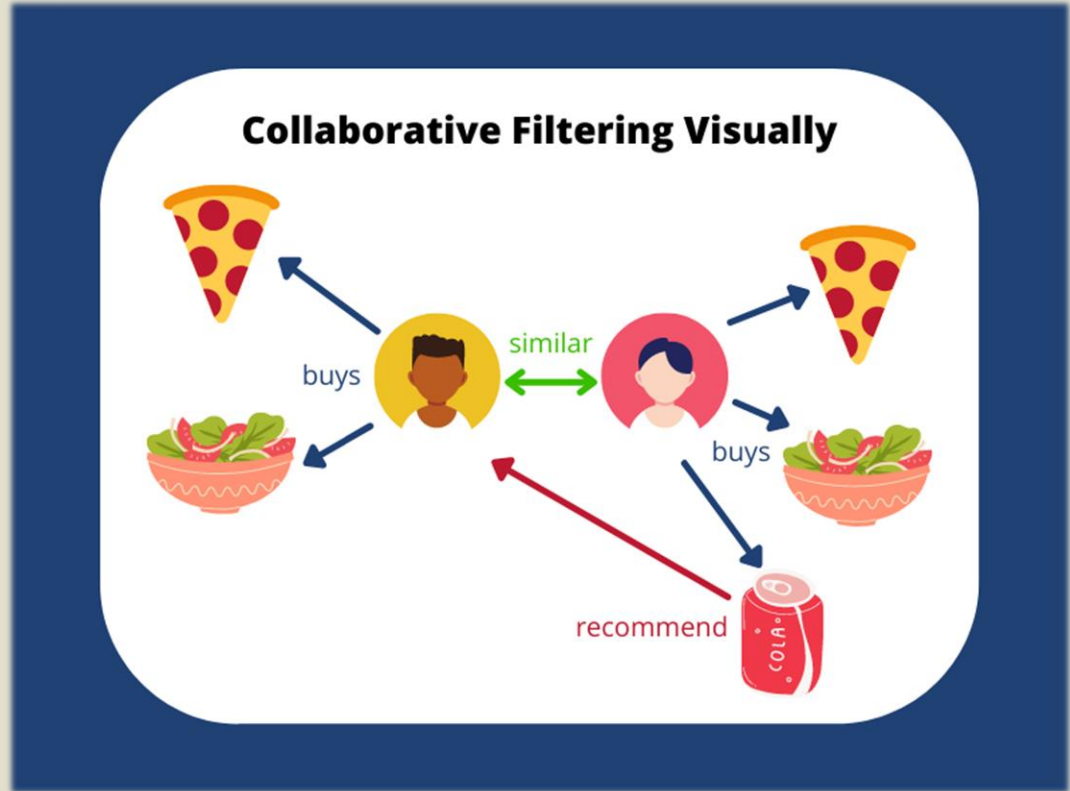


Types of recommendation

Collaborative-based Filtering  
Content-based Filtering

# What is Collaborative Filtering?

- Finds similar patterns among users
- Filters out items that users like based on the ratings or reactions of similar users



# Types of Collaborative Filtering



Based on User  
similarity

- Find users sharing similar rating patterns with test user
- Ratings from these users used to calculate a prediction



Based on item  
similarity

- Relationships among items found using item-item matrix
- Match user's data with this matrix to get preferences

## Similarity Check

Similarity can be computed as follows

$$r_{u,i} = \frac{1}{N} \sum_{u' \in U} r_{u',i}$$

$$r_{u,i} = k \sum_{u' \in U} \text{simil}(u, u') r_{u',i}$$

where  $k$  is a normalizing factor defined as  $k = 1 / \sum_{u' \in U} |\text{simil}(u, u')|$ , and

$$r_{u,i} = \bar{r}_u + k \sum_{u' \in U} \text{simil}(u, u') (r_{u',i} - \bar{r}_{u'})$$

[Source](#)

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

[Source](#)

# PyTorch



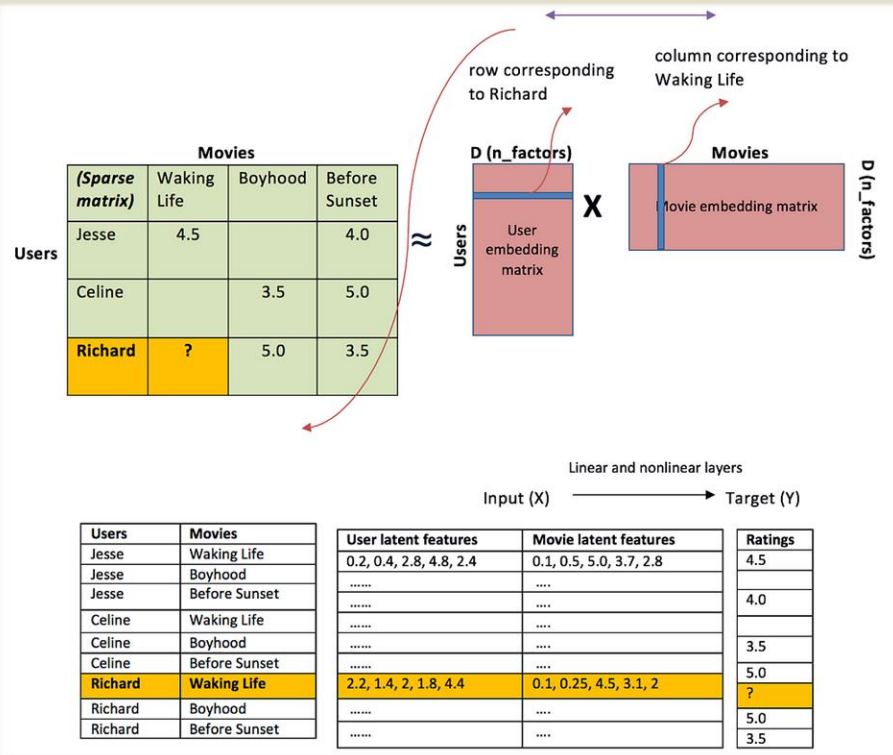
Based on an open-source machine learning library called Torch



Python frontend combined with Torch's powerful GPU-accelerated backend libraries to create a flexible and efficient system



Framework for building Deep Learning models



# Embeddings



The users and movies are converted into embeddings



The multiplication of these matrices gives a prediction



This prediction is then scaled to the range of the ratings

# Dimension Reduction Methods

1. t-Distributed Stochastic Neighbor Embedding (TSNE)
2. Principal component analysis-PCA
3. Uniform Manifold Approximation and Projection, UMAP

[Source](#)



# PyTorch Deep Neural Network Implementation



```
RecommendationDataset(  
    (u): Embedding(943, 50)  
    (m): Embedding(1682, 50)  
    (ub): Embedding(943, 1)  
    (mb): Embedding(1682, 1)  
    (lin1): Linear(in_features=50, out_features=2, bias=True)  
    (lin2): Linear(in_features=2, out_features=1, bias=True)  
    (drop1): Dropout(p=0.2, inplace=False)  
    (drop2): Dropout(p=0.2, inplace=False)  
)
```

# Model Summary

---

```
The prediction for the users tensor([1, 1, 2, 2, 3, 3, 4, 4]) and movies tensor([ 242,    1, 242, 1674, 242, 1674, 242, 1674]) are tensor([[3.1216],  
      [3.1214],  
      [3.0915],  
      [3.1183],  
      [3.1178],  
      [3.1167],  
      [3.1192],  
      [3.0901],  
      [3.1181],  
      [3.1193],  
      [3.1188],  
      [3.0905],  
      [3.1075],  
      [3.1175],  
      [3.0816],  
      [3.1176]], grad_fn=<AddBackward0>)
```

# Predictions

	0	1	2	3	4
196	Ed Wood (1994)	Under Siege (1992)	Brother Minister: The Assassination of Malcolm X (1994)	Long Kiss Goodnight, The (1996)	Rumble in the Bronx (1995)
63	Under Siege (1992)	Ed Wood (1994)	Long Kiss Goodnight, The (1996)	Brother Minister: The Assassination of Malcolm X (1994)	Rumble in the Bronx (1995)
226	Ed Wood (1994)	Under Siege (1992)	Brother Minister: The Assassination of Malcolm X (1994)	Long Kiss Goodnight, The (1996)	Rumble in the Bronx (1995)
154	Ed Wood (1994)	Brother Minister: The Assassination of Malcolm X (1994)	Under Siege (1992)	Long Kiss Goodnight, The (1996)	0
306	Ed Wood (1994)	Brother Minister: The Assassination of Malcolm X (1994)	Under Siege (1992)	Long Kiss Goodnight, The (1996)	Rumble in the Bronx (1995)
...	...	...	...	...	...
799	Ed Wood (1994)	Brother Minister: The Assassination of Malcolm X (1994)	Under Siege (1992)	Long Kiss Goodnight, The (1996)	0
358	Ed Wood (1994)	Brother Minister: The Assassination of Malcolm X (1994)	Under Siege (1992)	Long Kiss Goodnight, The (1996)	Rumble in the Bronx (1995)
410	Under Siege (1992)	Ed Wood (1994)	Long Kiss Goodnight, The (1996)	Rumble in the Bronx (1995)	Brother Minister: The Assassination of Malcolm X (1994)
598	Ed Wood (1994)	Brother Minister: The Assassination of Malcolm X (1994)	Under Siege (1992)	Long Kiss Goodnight, The (1996)	0
873	Ed Wood (1994)	Long Kiss Goodnight, The (1996)	Under Siege (1992)	Rumble in the Bronx (1995)	Brother Minister: The Assassination of Malcolm X (1994)

# Recommendation

# Fastai Implementation

A thin vertical line is positioned to the right of the text. At the bottom of the slide, there is a solid green horizontal bar.

# Collaborative filtering using fastai



What is fast ai?

Library built on PyTorch



Elements used to build a collaborative filter  
from fast ai



Data loader

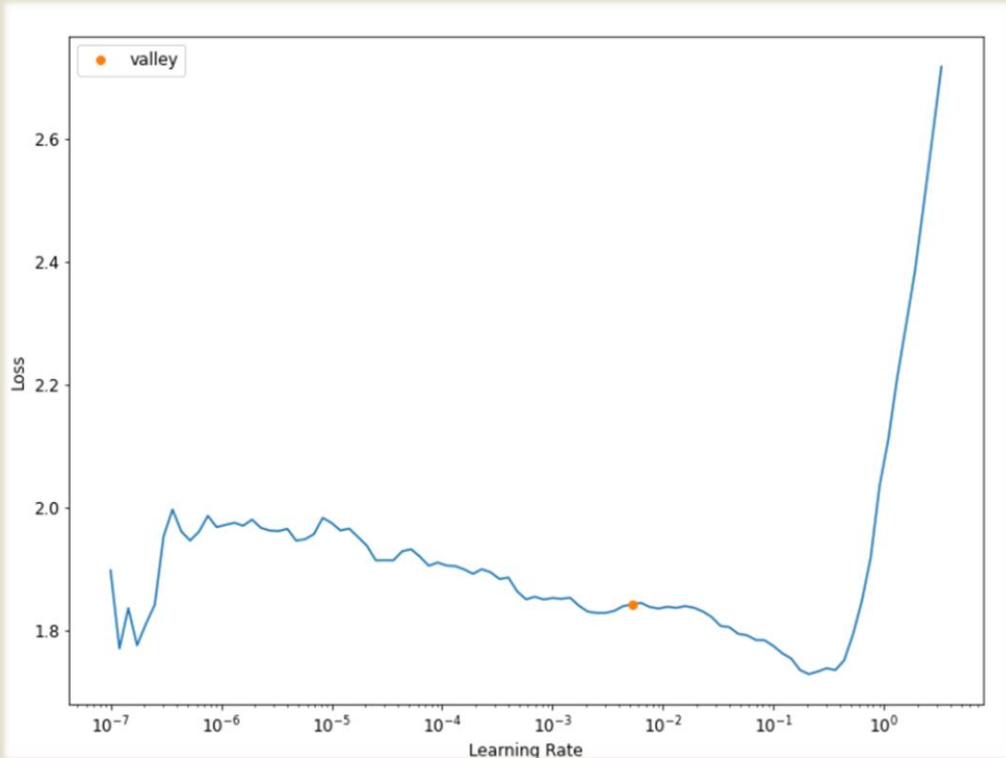


Learner



loss function(MSE)

# Choosing a Learning rate



## Model training

epoch	train_loss	valid_loss	time
0	0.950376	0.945305	00:13
1	0.842313	0.869440	00:13
2	0.744235	0.828205	00:13
3	0.580948	0.813320	00:13
4	0.471395	0.812804	00:13

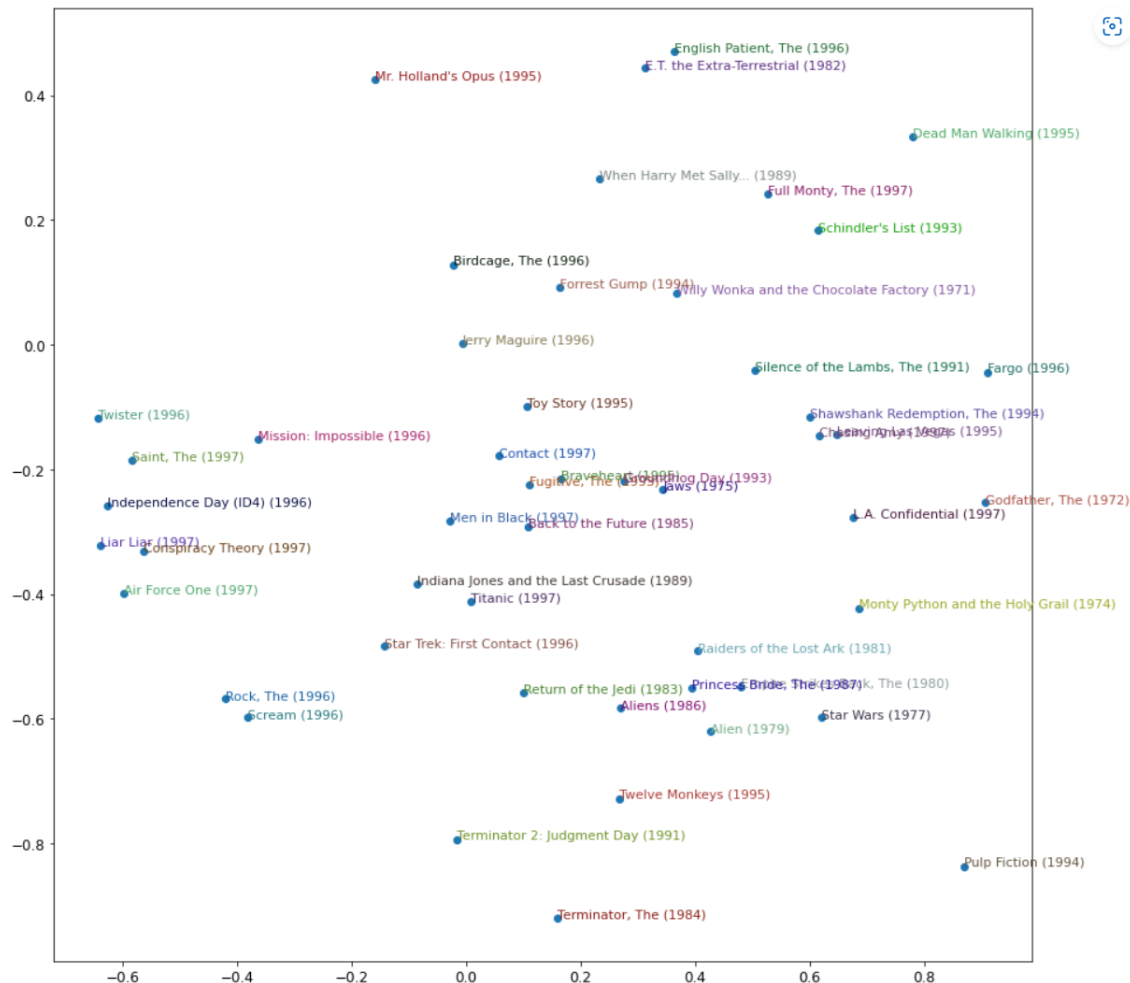


	user	movie	rating	title
0	1	6	5	Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)
1	9	6	5	Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)
2	63	6	3	Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)
3	79	6	4	Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)
4	90	6	4	Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)
5	409	6	4	Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)
6	1	10	3	Richard III (1995)
7	7	10	4	Richard III (1995)
8	49	10	3	Richard III (1995)
9	59	10	4	Richard III (1995)

	Prediction
0	4.172879
1	4.326862
2	3.140697
3	4.168423
4	4.267342
5	3.732576
6	3.729470
7	4.075462
8	4.010137
9	3.723830

Test predicted and actual value

# 2D Embedding



	0	1	2	3	4
196	Amadeus (1984)	Lone Star (1996)	Crash (1996)	Platoon (1986)	Hotel de Love (1996)
63	Two Much (1996)	Three Lives and Only One Death (1996)	Colonel Chabert, Le (1994)	Telling Lies in America (1997)	Young Guns (1988)
226	Brothers in Trouble (1995)	What Happened Was... (1994)	Jupiter's Wife (1994)	Colonel Chabert, Le (1994)	Senseless (1998)
154	Full Speed (1996)	Mallrats (1995)	Lone Star (1996)	Serial Mom (1994)	Thin Man, The (1934)
306	Heathers (1989)	Young Poisoner's Handbook, The (1995)	Spice World (1997)	Telling Lies in America (1997)	Manny & Lo (1996)
...	...	...	...	...	...
799	Full Speed (1996)	My Favorite Year (1982)	U.S. Marshalls (1998)	Sleeper (1973)	Colonel Chabert, Le (1994)
358	Basquiat (1996)	Kalifornia (1993)	Mark of Zorro, The (1940)	When We Were Kings (1996)	Super Mario Bros. (1993)
410	Full Speed (1996)	Colonel Chabert, Le (1994)	Senseless (1998)	Heathers (1989)	What Happened Was... (1994)
598	Mallrats (1995)	Blue in the Face (1995)	Turbulence (1997)	Seven Years in Tibet (1997)	Philadelphia Story, The (1940)
873	Madonna: Truth or Dare (1991)	Big Bang Theory, The (1994)	Late Bloomers (1996)	Nightwatch (1997)	Ulee's Gold (1997)

---

# Recommendation

# Future scope

- § Incorporate into real world applications
- § Test for a larger data set
- § Get the likings of the user initially as that data is learnt over time

# Sources

1. [fastai - Collaborative filtering tutorial](#)
2. <https://pytorch.org/docs/stable/index.html>
3. <https://ai.plainenglish.io/fast-ai-recommendations-using-collaborative-filtering-d2dec7c702e9>
4. [Interactive Analysis of Sentence Embeddings \(amitnness.com\)](#)
5. [Neural Network Embeddings Explained | by Will Koehrsen | Towards Data Science](#)
6. F. Maxwell Harper and Joseph A. Konstan. 2015  
[The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages. DOI=<http://dx.doi.org/10.1145/2827872>]