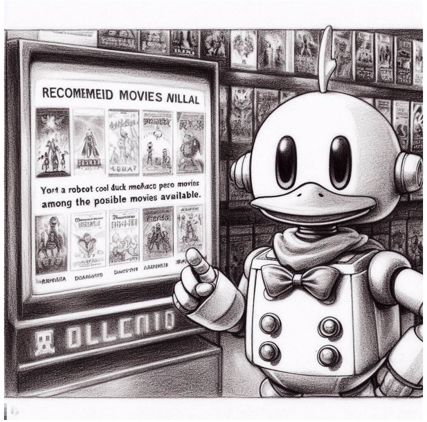
Introduction to recommender systems



AI frameworks

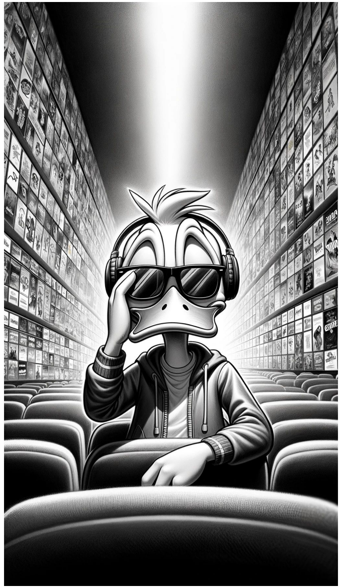
Course: 6 videos

- Introduction
- Popularity and content based methods
- Collaborative filtering
- Matrix factorization
- Neural collaborative filtering
- Evaluating recommender systems

Introduction

Objectives

- Help users to match with the best items
- Ease information overload


Objectives:

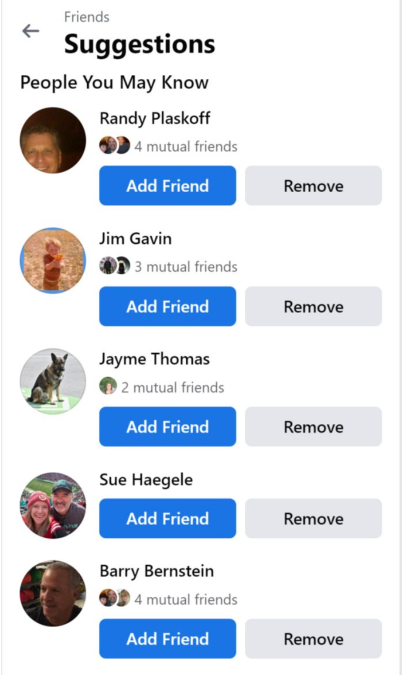
- Information filtering able to predict appetency.

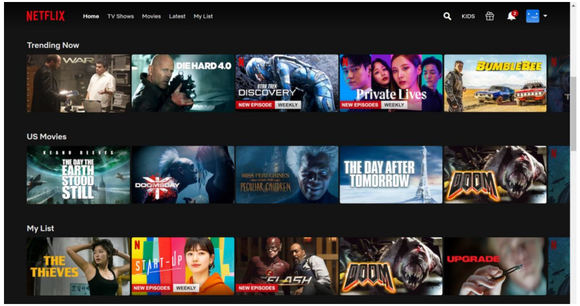
Objectives:

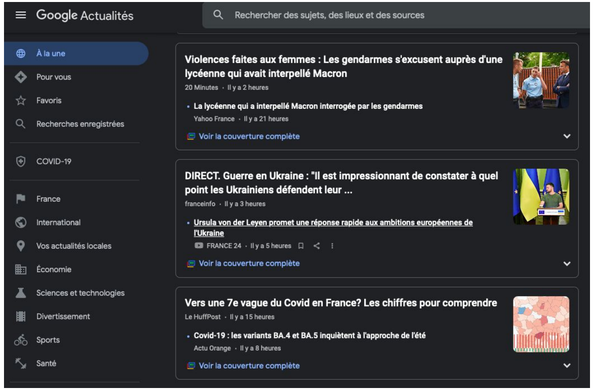
- Information filtering able to predict appetency.
- Different of search $\neq$

Applications:

- E-commerce
- Media streaming services
- News and content aggregation websites
- Social netwoks







Applications:

- Netflix => 2/3 of the movies watched are recommended
- Google =>news recommendations improved click-through rate (CTR) by 38%
- Amazon => 35% of sales come from recommendations

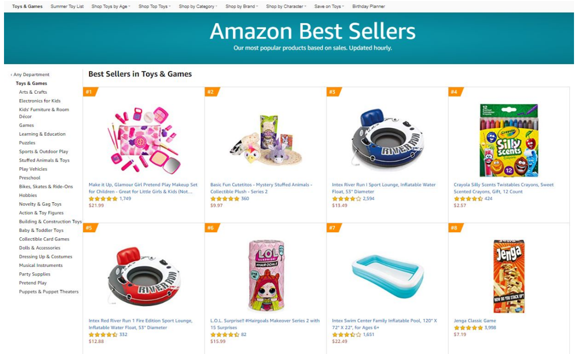
History and evolution

- Rule based systems
- Xerox Palo Alto Research => Tapestry 1992 collaborative filtering Mails
- GroupLens system => 1992 Matrix filling problem News
- Netflix prize 2006-2009
- Lots of side information (item-related data, context, social-networks,...)
- Sequence-aware recommendations

Types of recommender systems

Types of recommender systems

- Popularity

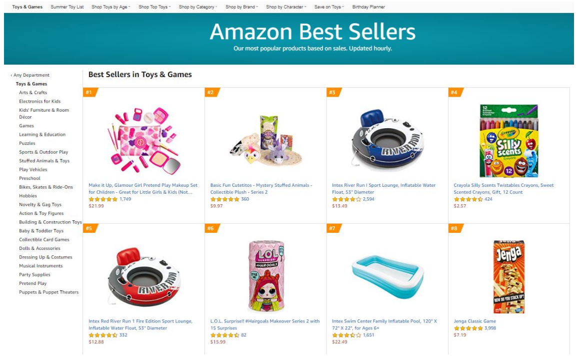


Types of recommender systems

- Popularity
- Content based

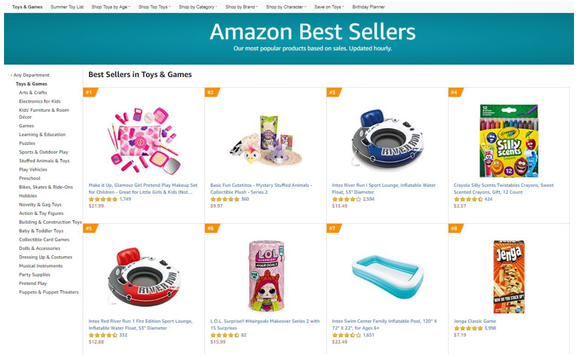
Hopefully you love the Unearth Women brand as much as we have loved creating it. Our mission has always been to support women, lift their voices and, simply put, *unearth* women's stories. This female-designed ceramic coffee mug features the Unearth Women logo printed across the mug. The mug measures at a height of 3.85" (9.8 cm) and diameter of 3.35" (8.5 cm) and is microwave and dishwasher safe.

As always, every purchase from the Unearth Women store helps support our mission to pay our female content creators fairly and keep our platform growing.



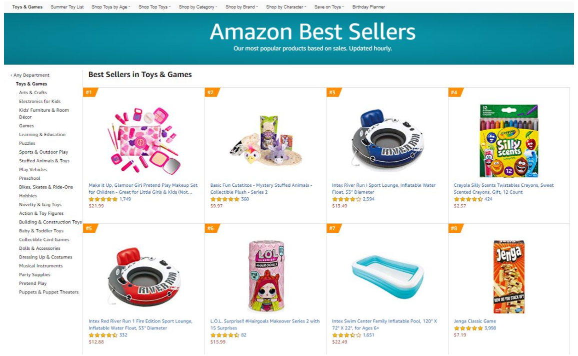
Types of recommender systems

- Popularity
- Content based
- Collaborative filtering



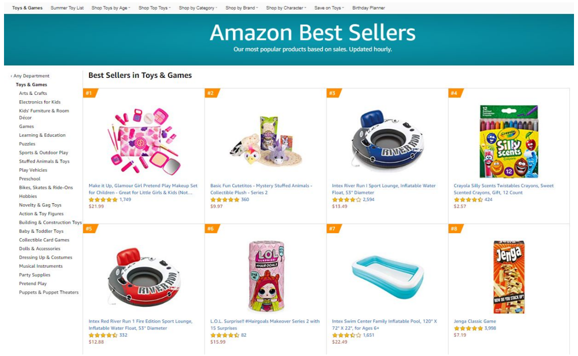
Types of recommender systems

- Popularity
- Content based
- Collaborative filtering
- Graph-based

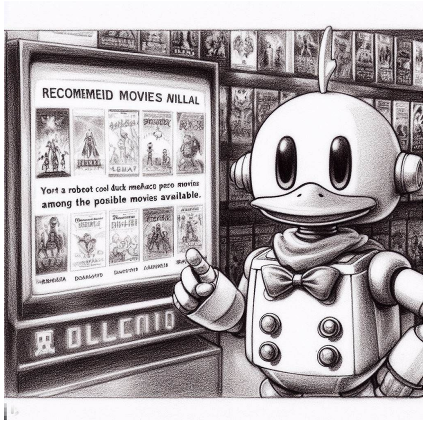


Types of recommender systems

- Popularity
- Content based
- Collaborative filtering
- Graph-based
- Hybrid



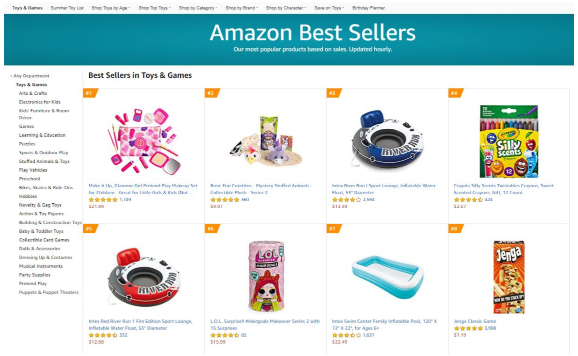
Introduction to recommender systems

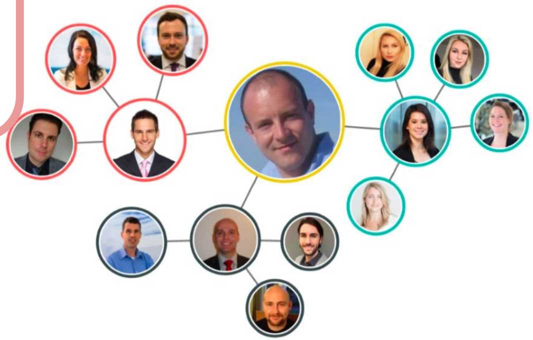


AI frameworks

Types of recommender systems

- Popularity
- Content based
- Collaborative filtering
- Graph-based
- Hybrid

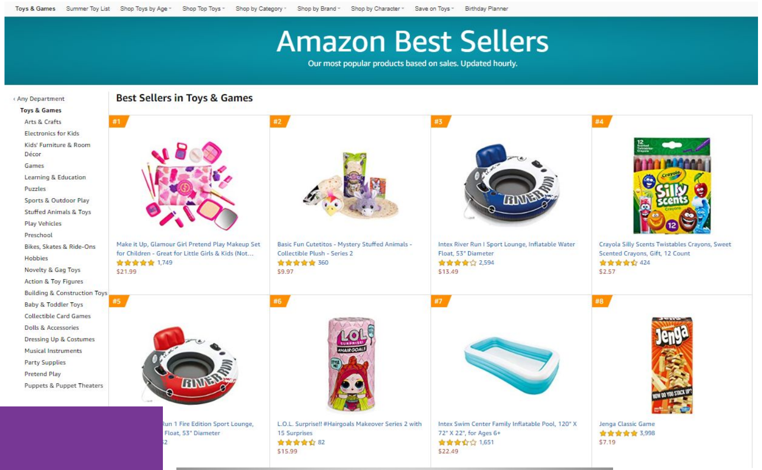


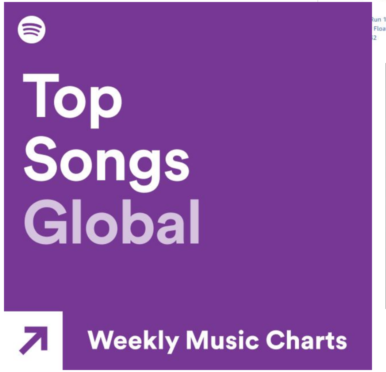


Popularity based recommendations

Popularity based:

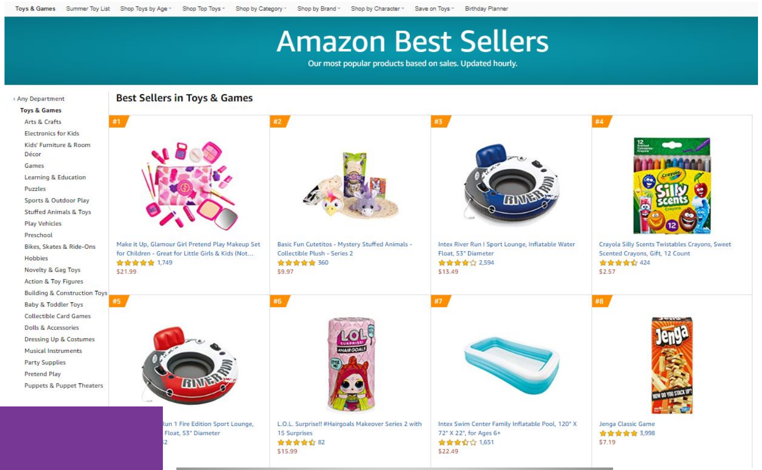
- Rank items and recommend most popular

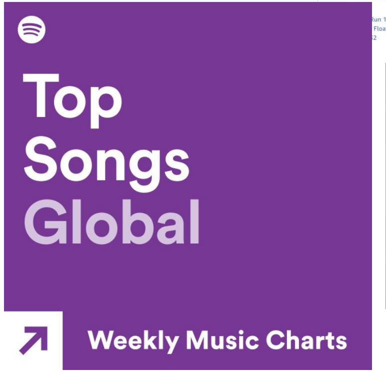




Popularity based:

- Rank items and recommend most popular
- Can be specified by category or time period



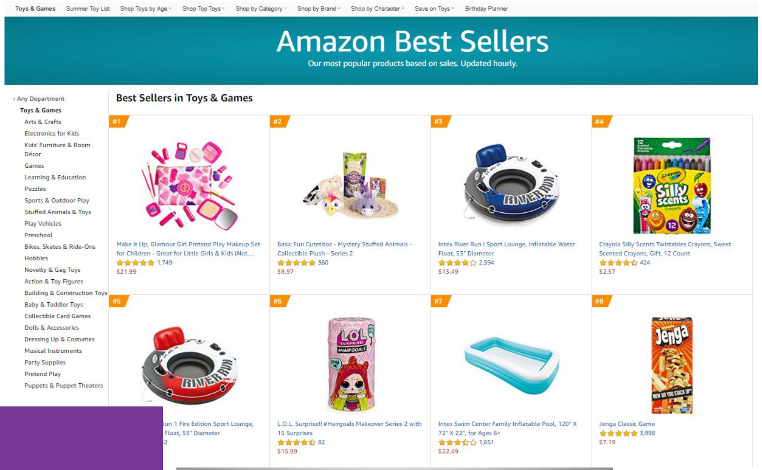


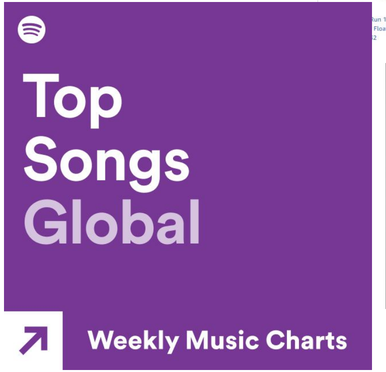
Popularity based:

- Rank items and recommend most popular
- Can be specified by category or time period

Advantages:

- Easy to set up
- No information on users needed
- Good cold start
- Good baseline





Popularity based:

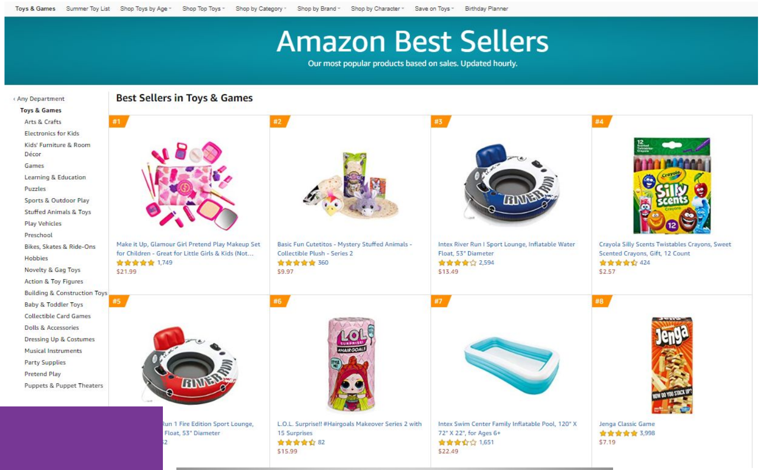
- Rank items and recommend most popular
- Can be specified by category or time period

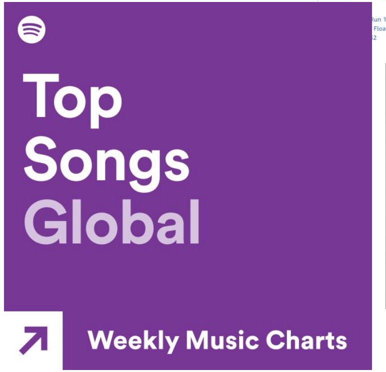
Advantages:

- Easy to set up
- No information on users needed
- Good cold start
- Good baseline

Drawbacks:

- No personalization
- Popular items may end up dominating the recommendations





Content based recommendations

Content based

Goal: Find most similar items based on their characteristics

Content based

Item description:

Marque	Xinlon
Dimensions du produit	7,5L x 6l x 6H centimètres
Tranche d'âge (description)	Enfant
Nombre de pièces	2
Nombre d'articles	2
Nom de la collection	Animal
Nombre d'unités	2 Count
Fabricant	Xinlongyi-Bin

2 x Rubber ducks

Content based

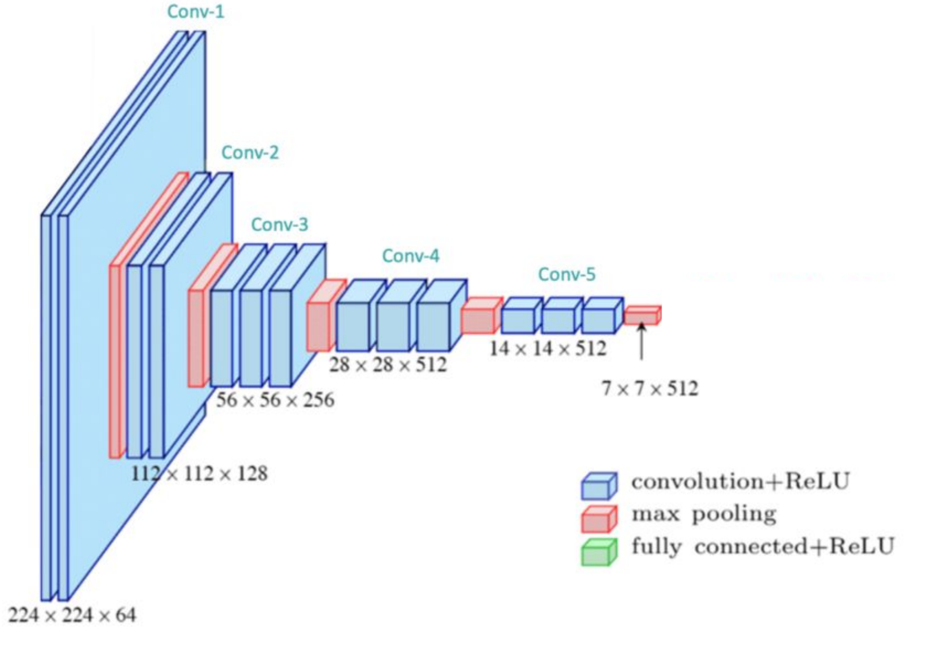
Textual description:

À propos de cet article

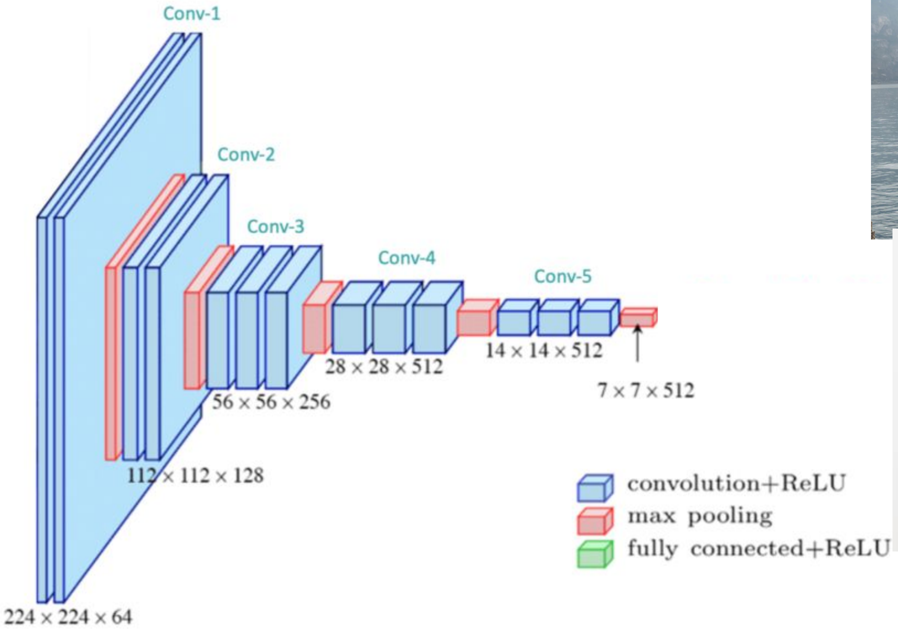
- **Matériaux de haute qualité**: les jouets de canard jaune sont fabriqués en silicone sans danger et ne causeront aucun dommage à la peau humaine. Ils ont une surface lisse et un bon toucher, lisses, doux, résistants aux températures élevées et difficiles à décolorer.
- **Beau design**: la décoration de la voiture de canard jaune est équipée de lunettes de soleil, de colliers et de casques sympas. La forme est très intéressante et mignonne, et il est facile d'attirer l'attention des autres. Il convient à la décoration de votre voiture et de votre casque.
- **Assemblage sans bricolage**: Le jouet mignon de canard est livré avec des casques intéressants, des lunettes de soleil, des colliers et du ruban adhésif double face. Vous devez l'assembler vous-même. Vous pouvez utiliser votre imagination pour vous habiller comme vous le souhaitez. Le casque de ce canard est détachable.
- **Contenu de l'emballage**: Le produit comprend 2 canards en caoutchouc, 2 casques de styles différents, 2 hélices rotatives, 2 lunettes de soleil, 2 colliers et 2 autocollants double face super collants.
- **Scène applicable**: ce type de canard de voiture peut également être collé sur des chariots et des vélos pour enfants. Lorsque les enfants voient des canards jaunes sur des poussettes et des vélos et regardent les hélices sur leurs casques tourner, ils se sentiront très heureux et obéissants.

Content based

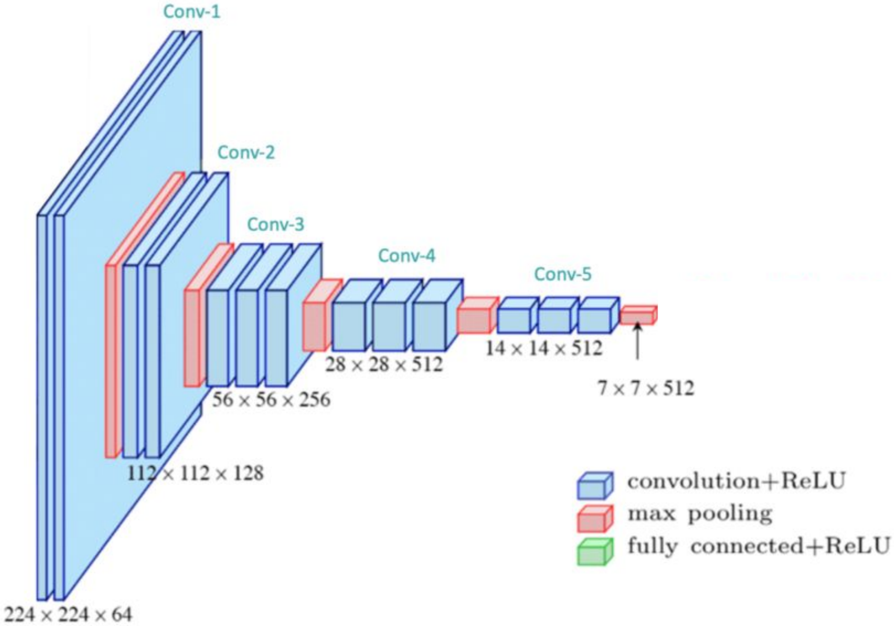
Visual description:

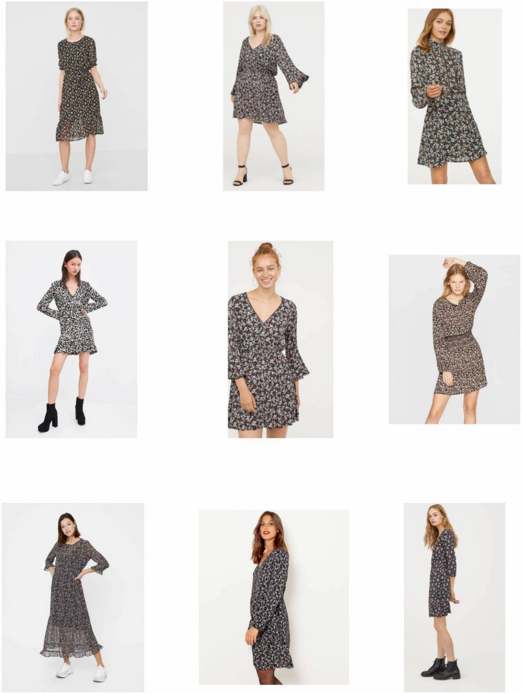


Content based



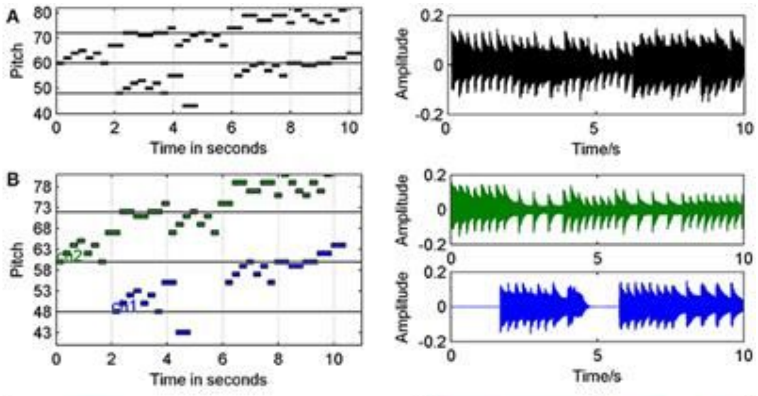
Content based





Content based

Audio description:



Similarity measures

Similarity measures

Embedding space E

E

[3.2, 1.5, ..., -2.1]

[2.8, 3.5, ..., 0.3]

● ● ●

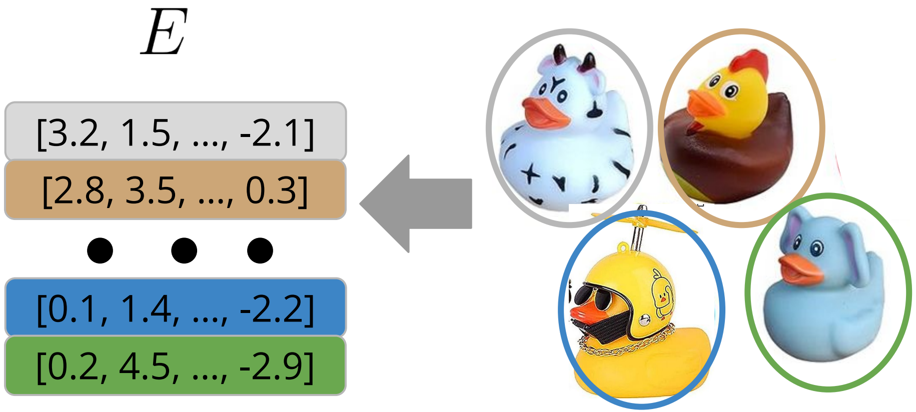
[0.1, 1.4, ..., -2.2]

[0.2, 4.5, ..., -2.9]

Similarity measures

Embedding space E

Similarity measure $s : E \times E \to \mathbb{R}$



Similarity measures

Embedding space E

Similarity measure $s : E \times E \to \mathbb{R}$

Given a **query** $q \in E$

[0.2, 1.5, ..., -2.1]

E

[3.2, 1.5, ..., -2.1]

[2.8, 3.5, ..., 0.3]

● ● ●

[0.1, 1.4, ..., -2.2]

[0.2, 4.5, ..., -2.9]

Similarity measures

Embedding space E

Similarity measure $s : E \times E \to \mathbb{R}$

Given a **query** $q \in E$ the system looks for item embeddings $x \in E$ with **high similarity** $s(q, x)$

[0.2, 1.5, ..., -2.1]

E

[3.2, 1.5, ..., -2.1]

[2.8, 3.5, ..., 0.3]

● ● ●

[0.1, 1.4, ..., -2.2]

[0.2, 4.5, ..., -2.9]

Similarity measures

Embedding space E

Similarity measure $s : E \times E \to \mathbb{R}$

Given a **query** $q \in E$ the system looks for item embeddings $x \in E$ with **high similarity** $s(q, x)$

- Cosine similarity $s(q, x) = cos(q, x)$

Similarity measures

Embedding space E

Similarity measure $s : E \times E \rightarrow \mathbb{R}$

Given a **query** $q \in E$ the system looks for item embeddings $x \in E$ with **high similarity** $s(q, x)$

- Cosine similarity $s(q, x) = cos(q, x)$
- Dot Product $s(q, x) = \langle q, x \rangle = \sum_{i=1}^{d} q_i x_i = \|x\| \|q\| \cos(q, x)$

Similarity measures

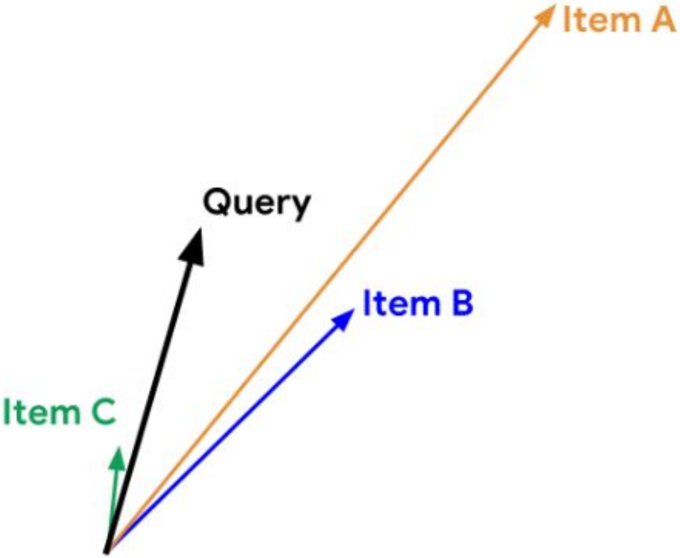
Embedding space E

Similarity measure $s : E \times E \to \mathbb{R}$

Given a **query** $q \in E$ the system looks for item embeddings $x \in E$ with **high similarity** $s(q, x)$

- Cosine similarity $s(q, x) = cos(q, x)$
- Dot Product $s(q, x) = \langle q, x \rangle = \sum_{i=1}^{d} q_i x_i = \|x\| \|q\| \cos(q, x)$
- Euclidian distance $s(q, x) = \|q - x\| = \left[\sum_{i=1}^{d} (q_i - x_i)^2\right]^{\frac{1}{2}}$

Similarity



Dot-Product

Query : Item A > Item B > Item C

Cosine

Query : Item C > Item A > Item B

(-) Euclidean Distance

Query : Item B > Item C > Item A

Source: https://developers.google.com/machine-learning/recommendation

Content based

Advantages:

- No need of user interactions
- Good for cold start
- Easy to scale to large number of users
- Easy to use with new product

Content based

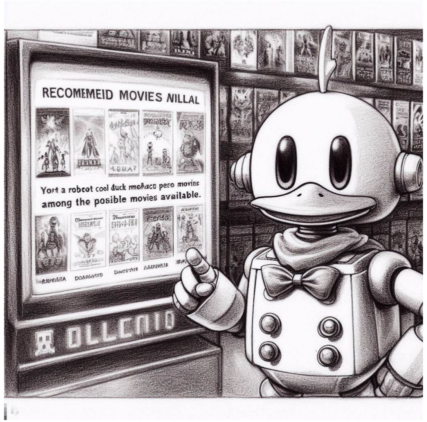
Advantages:

- No need of user interactions
- Good for cold start
- Easy to scale to large number of users
- Easy to use with new product

Drawbacks:

- Hard to extract relevant embeddings
- Embeddings are often handcrafted and require expert knowledge

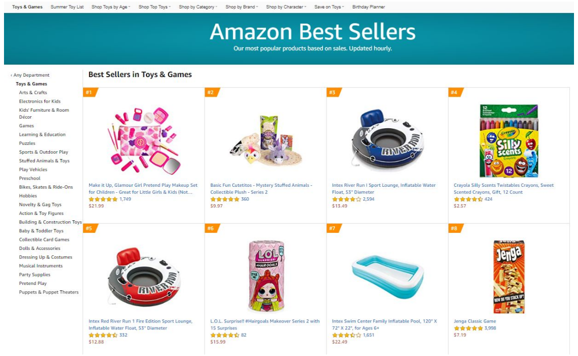
Introduction to recommender systems

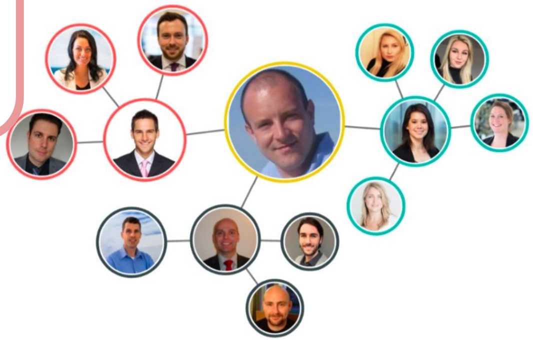


AI frameworks

Types of recommender systems

- Popularity
- Content based
- Collaborative filtering
- Graph-based
- Hybrid

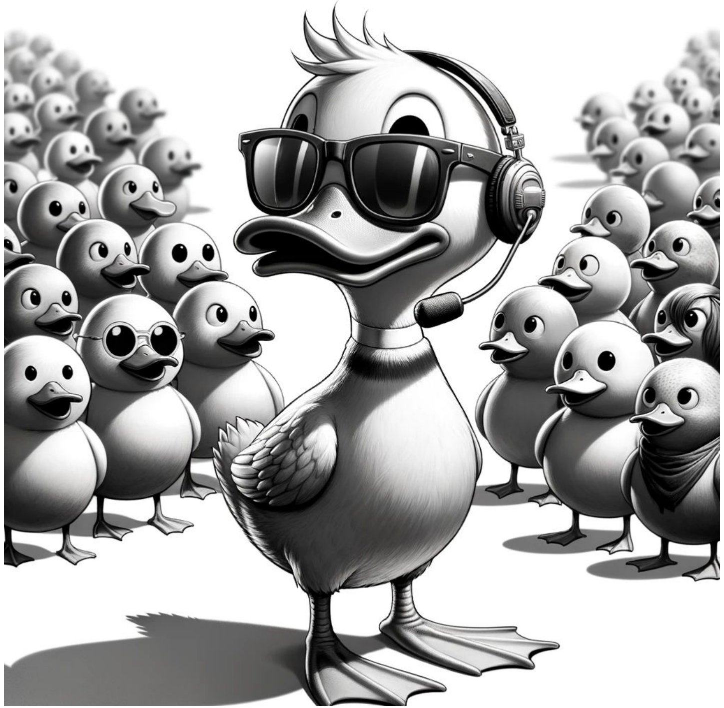




Collaborative filtering

Collaborative filtering

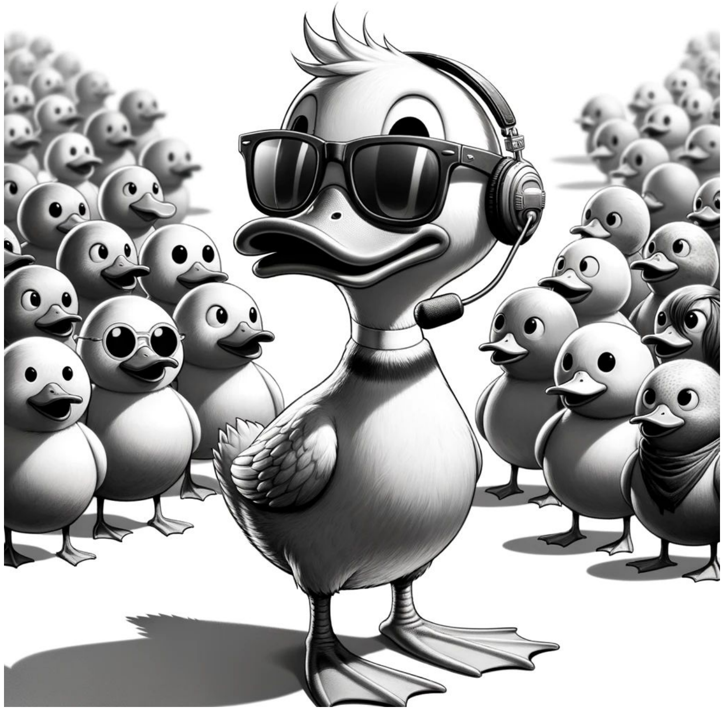
Goal:
Predict the interests of a user using preferences from other users.



Collaborative filtering

Goal:
Predict the interests of a user using preferences from other users.

- Users x Items interactions
- User based
- Item based
- Matrix factorization



Collaborative filtering

Based on interactions
User x Items

Two strategies:

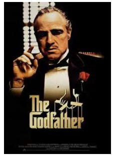
- User-based
- Item-based

					i
u	4	1	4	3	$r_{u,i}$
	1	5	5	4	4
	2	2	2	3	2
	5	5	1	1	1
	4	2	4	3	4
	3	1	4	3	3

User based

Principle:

- Find most similar users
- Estimate rating by the weighted average of similar users

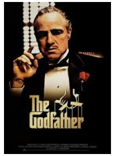
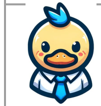
					i 
 u	4	1	4	3	$\boxed{r_{u,i}}$
	1	5	5	4	4
	2	2	2	3	2
	5	5	1	1	1
	4	2	4	3	4
	3	1	4	3	3

User based

Similar users:

- k-nearest neighbors
- Pearson correlation
- Cosine similarity

Neighborhood U

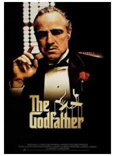
					i 
 u	4	1	4	3	$\boxed{r_{u,i}}$
	1	5	5	4	4
	2	2	2	3	2
	5	5	1	1	1
	4	2	4	3	4
	3	1	4	3	3

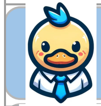
User based

Similar users:

- k-nearest neighbors
- Pearson correlation
- Cosine similarity

Neighborhood U




					i
 u	4	1	4	3	$\boxed{r_{u,i}}$
	1	5	5	4	4
	2	2	2	3	2
	5	5	1	1	1
	4	2	4	3	4
	3	1	4	3	3

User based

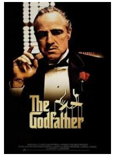
Similar users:

- k-nearest neighbors
- Pearson correlation
- Cosine similarity

Neighborhood U

Estimate rating $r_{u,i}$ **:**

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

					i 
 u	4	1	4	3	$\boxed{r_{u,i}}$
	1	5	5	4	4
	2	2	2	3	2
	5	5	1	1	1
	4	2	4	3	4
	3	1	4	3	3

User based

Similar users:

- k-nearest neighbors
- Pearson correlation
- Cosine similarity

Neighborhood U

Similarity measure $sim(u, u')$

Estimate rating $r_{u,i}$ **:**

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

with $$k = 1 / \sum_{u' \in U} |\operatorname{sim}(u, u')|$$

					i
u	4	1	4	3	$r_{u,i}$
	1	5	5	4	4
	2	2	2	3	2
	5	5	1	1	1
	4	2	4	3	4
	3	1	4	3	3

User based

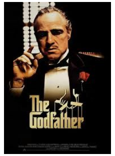
Similar users:

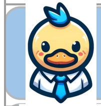
- k-nearest neighbors
- Pearson correlation
- Cosine similarity

Neighborhood U

Similarity measure $sim(u, u')$

Estimate rating $r_{u,i}$ **:**




					i
 u	4	1	4	3	$r_{u,i}$
	1	5	5	4	4
	2	2	2	3	2
	5	5	1	1	1
	4	2	4	3	4
	3	1	4	3	3

$r_{u,i} = \overline{r_u} + k \sum_{u' \in U} \mathrm{sim}\,(u, u')\,(r_{u',i} - \overline{r_{u'}})$ with $\overline{r_u}$ the average rating of user u

User based

u	4	1	4	3	$r_{u,i}$
	1	5	5	4	4
	2	2	2	3	2
	5	5	1	1	1
	4	2	4	3	4
	3	1	4	3	3

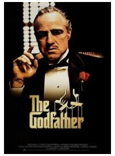
Drawbacks:

- How to handle new users?
- Does not scale to large real-world scenarios where |Users| >> |Items|

Item based

Principle:

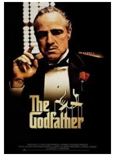
- Use similarity between items
- Recommend items similar with the current item or the user history

					i 
 u	4	1	4	3	$\boxed{r_{u,i}}$
	1	5	5	4	4
	2	2	2	3	2
	5	5	1	1	1
	4	2	4	3	4
	3	1	4	3	3

Item based

Similar items:

- k-nearest neighbors
- Cosine similarity

					i 
 u	4	1	4	3	$r_{u,i}$
	1	5	5	4	4
	2	2	2	3	2
	5	5	1	1	1
	4	2	4	3	4
	3	1	4	3	3

Item based

Similar items:

- k-nearest neighbors
- Cosine similarity

Neighborhood I

Similarity measure $sim(i, i')$

Estimate rating $r_{u,i}$ **:**

$$r_{u,i} = k \sum_{i' \in I} \operatorname{sim}(i, i')\, r_{u,i'}$$

with $k = 1 / \sum_{i' \in I} |\operatorname{sim}(i, i')|$

					i
u	4	1	4	3	$r_{u,i}$
	1	5	5	4	4
	2	2	2	3	2
	5	5	1	1	1
	4	2	4	3	4
	3	1	4	3	3

Item based

Advantages:

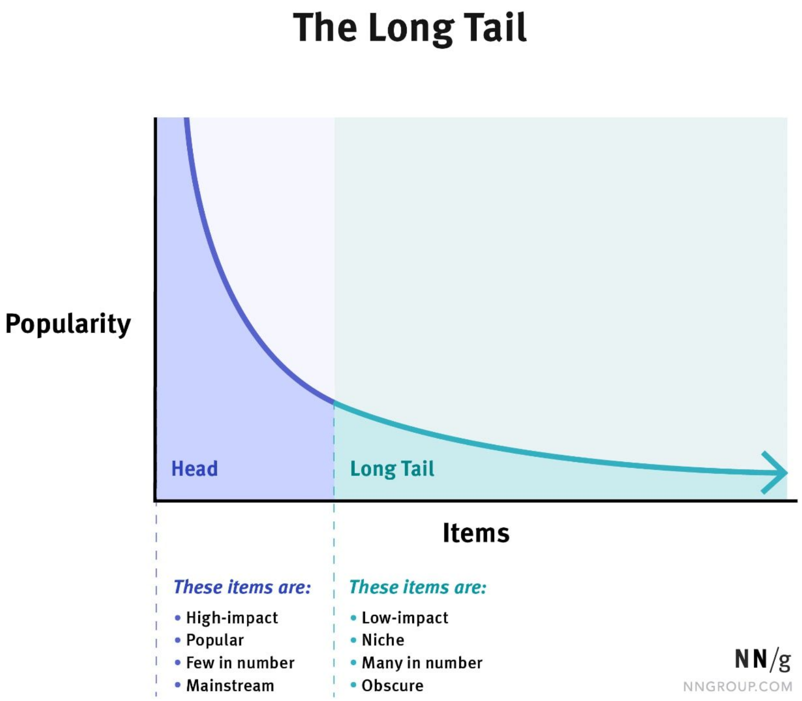
- Supposed to be more stable
- Pre-compute pairwise similarities
- Easier to scale

Drawbacks:

- New items
- Items with few interactions

					i
u	4	1	4	3	$r_{u,i}$
	1	5	5	4	2
	2	2	2	3	2
	5	5	1	1	4
	4	2	4	3	?
	3	1	4	3	5

Collaborative filtering

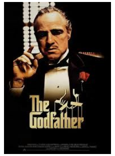
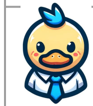


Source: https://www.nngroup.com/articles/long-tail/

Explicit vs Implicit Feedbacks

Explicit feedbacks:

- User rate items

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

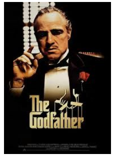
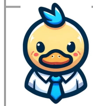
Explicit vs Implicit Feedbacks

Explicit feedbacks:

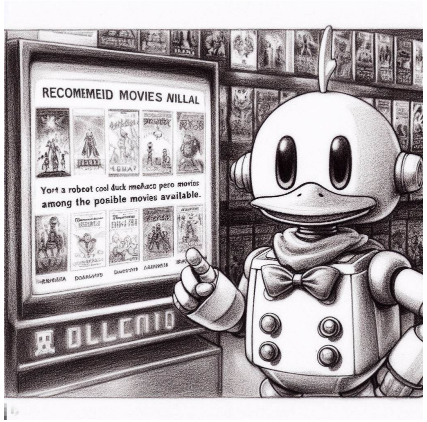
- User rate items

Implicit feedbacks:

- Clics, watched, bought, …
- Watch full video, …
- Stop watching, …

					
 u	1	1			1
		1			
			1		
	1		1	1	
	1				1
	1	1			

Introduction to recommender systems

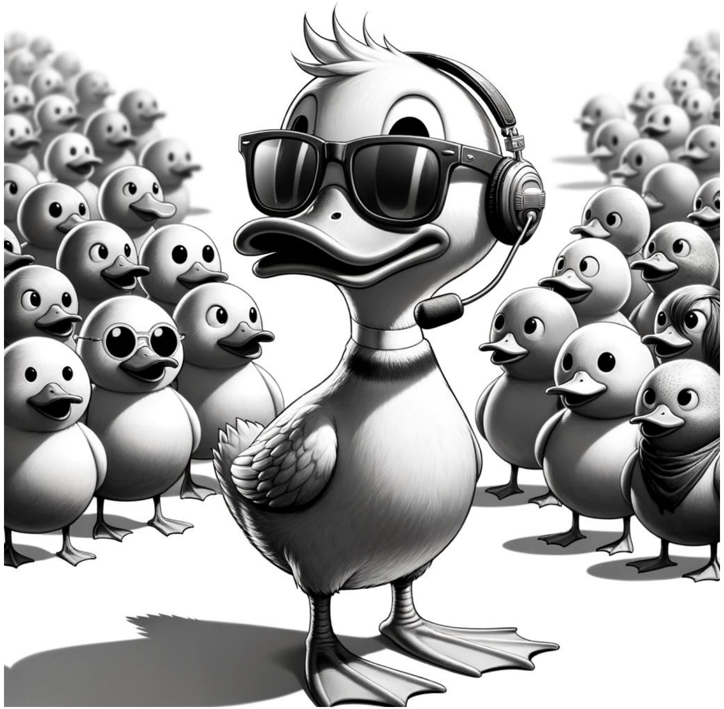


AI frameworks

Collaborative filtering

Goal:
Predict the interests of a user using preferences from other users.

- Users x Items interactions
- User based
- Item based
- Matrix factorization



Matrix factorization

Goal:

Factorize the interaction matrix A

A

Source: https://developers.google.com/machine-learning/recommendation

Matrix factorization

Goal:

Factorize the interaction matrix A

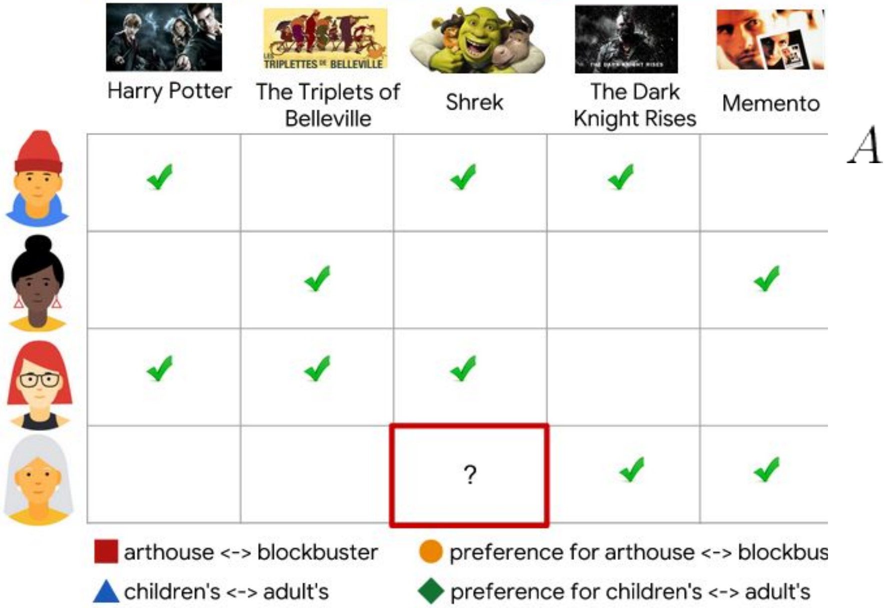
A

Source: https://developers.google.com/machine-learning/recommendation

Matrix factorization

Goal:

Factorize the interaction matrix A



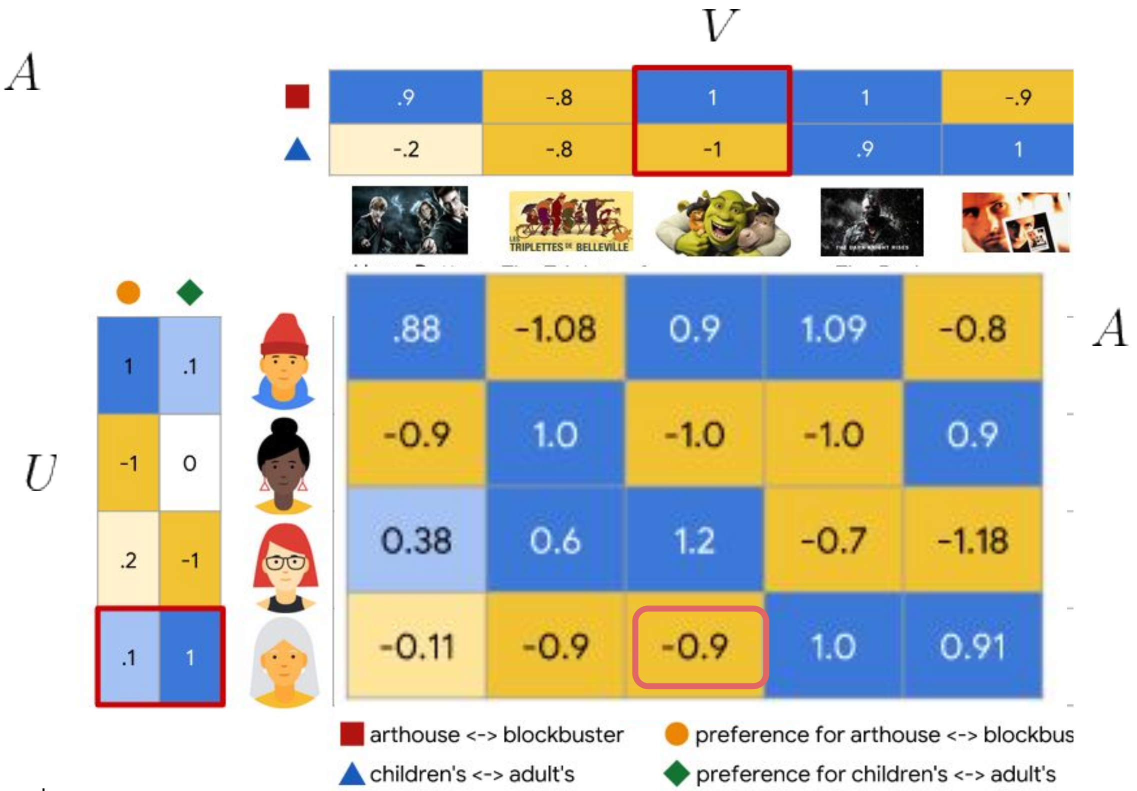
Source: https://developers.google.com/machine-learning/recommendation

Matrix factorization

Goal:

Factorize the interaction matrix A

- A user embedding matrix U
- An item embedding matrix V



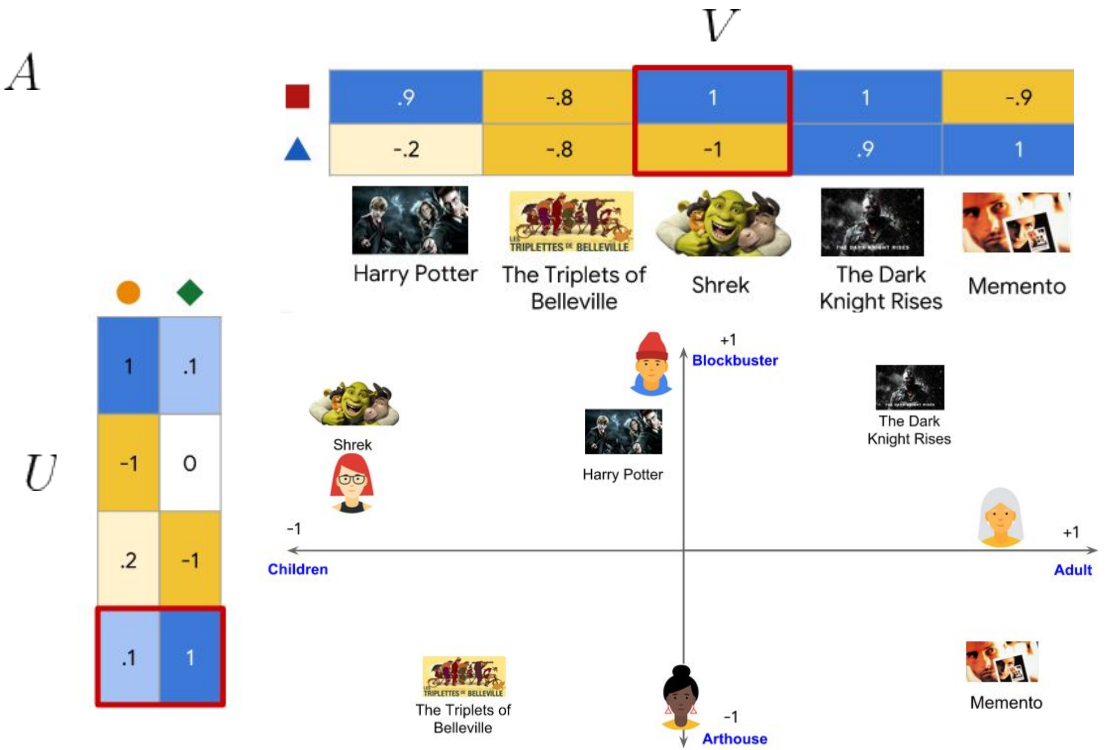
Source: https://developers.google.com/machine-learning/recommenda...

Matrix factorization

Goal:

Factorize the interaction matrix A

- A user embedding matrix U
- An item embedding matrix V



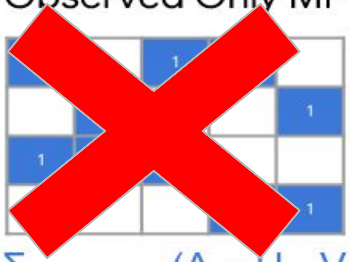
Source: https://developers.google.com/machine-learning/recommenda

Matrix factorization

Optimization:

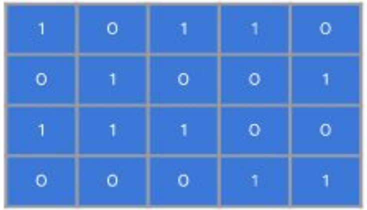
- Stochastic gradient descent
- Alternating Least Squares:
  - Fix U and solve for V
  - Fix V and solve for U

Observed Only MF



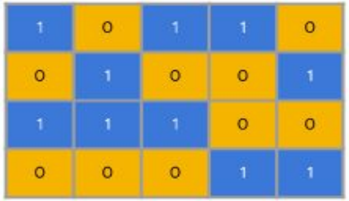
$\Sigma_{(i,\,j)\,\in\,\text{obs}}\ (A_{ij} - U_i \,.\, V_j)^2$

SVD



$|A - U V^T|_F^2$
$= \Sigma_{(i,\,j)}\ (A_{ij} - U_i \,.\, V_j)^2$

Weighted MF



$\Sigma_{(i,\,j)\,\in\,\text{obs}}\ (A_{ij} - U_i \,.\, V_j)^2 +$
$w_0\ \Sigma_{(i,\,j)\,\notin\,\text{obs}}\ (0 - U_i \,.\, V_j)^2$

Source: https://developers.google.com/machine-learning/recommendation

Matrix factorization

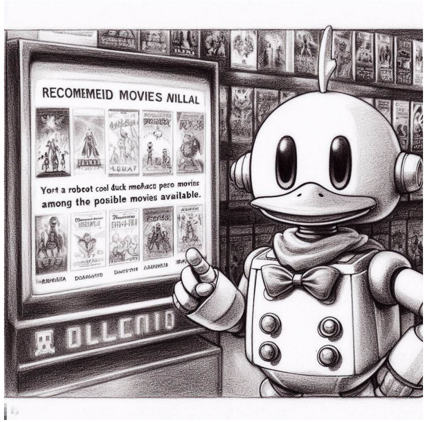
Advantages:

- Can be parallelized (ALS)
- Can be computed offline
- Embeddings can be used for item-item recommendations
- Good for serendipity

Drawbacks:

- Can’t handle new items
- Does not include other possible meaningful features

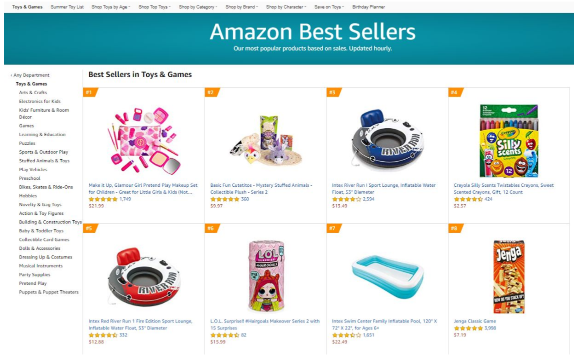
Introduction to recommender systems

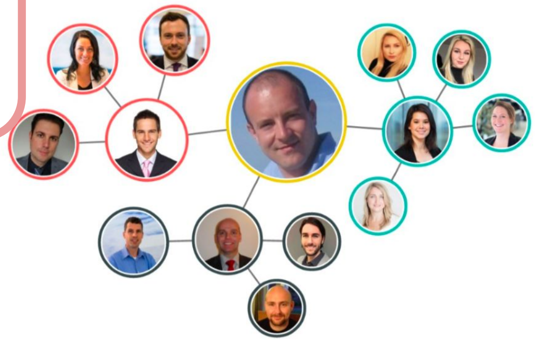


AI frameworks

Types of recommender systems

- Popularity
- Content based
- Collaborative filtering
- Graph-based
- Hybrid



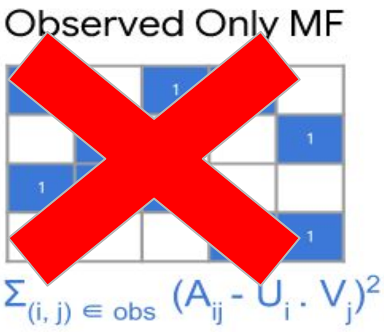


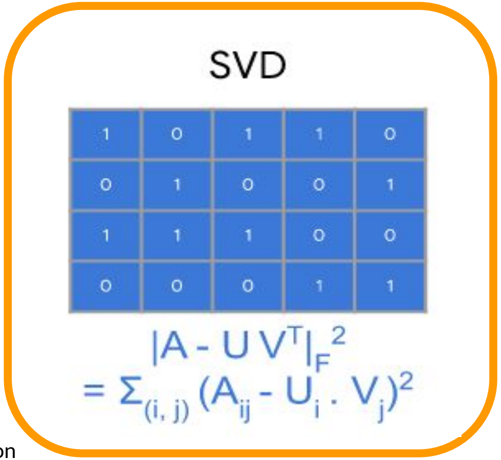
Neural collaborative systems

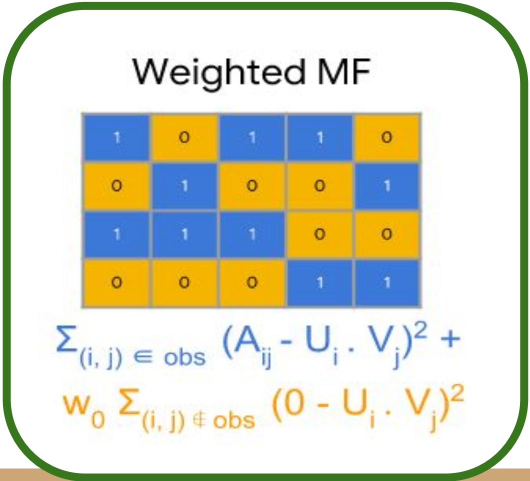
Matrix factorization

Optimization:

- Stochastic gradient descent
- Alternating Least Squares:
  - Fix U and solve for V
  - Fix V and solve for U

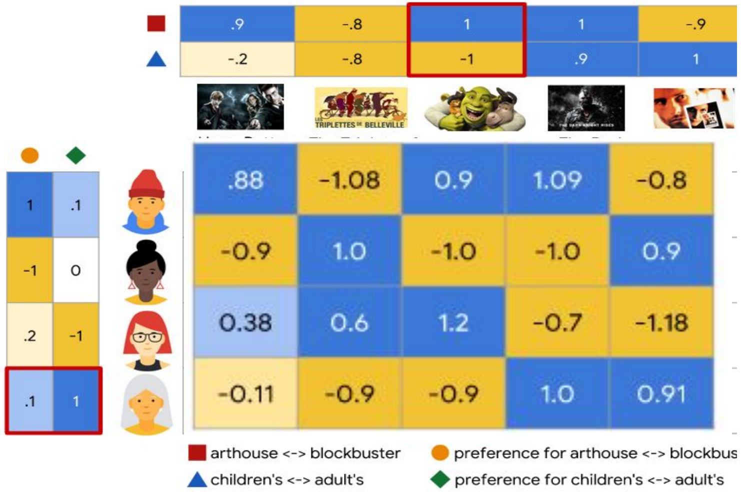


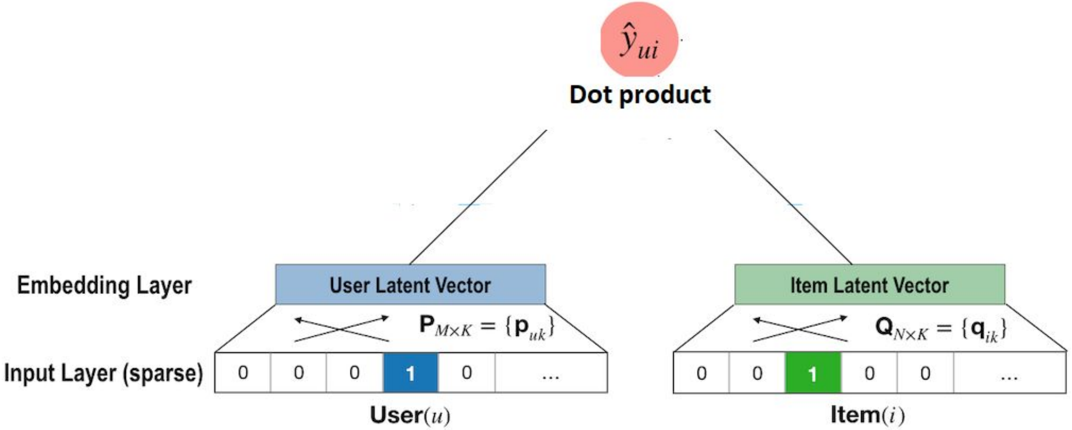




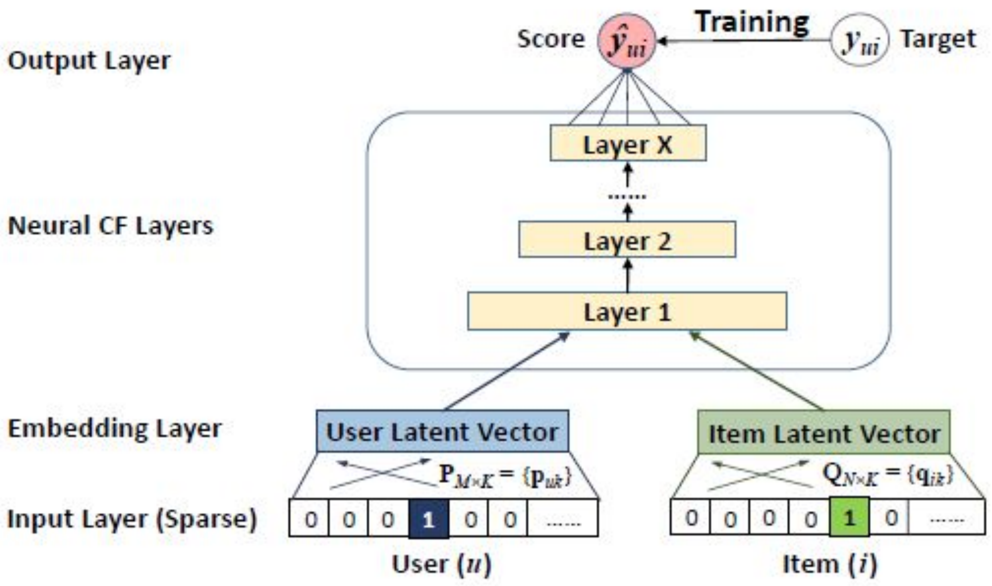
Source: https://developers.google.com/machine-learning/recommendation

Neural Collaborative Filtering



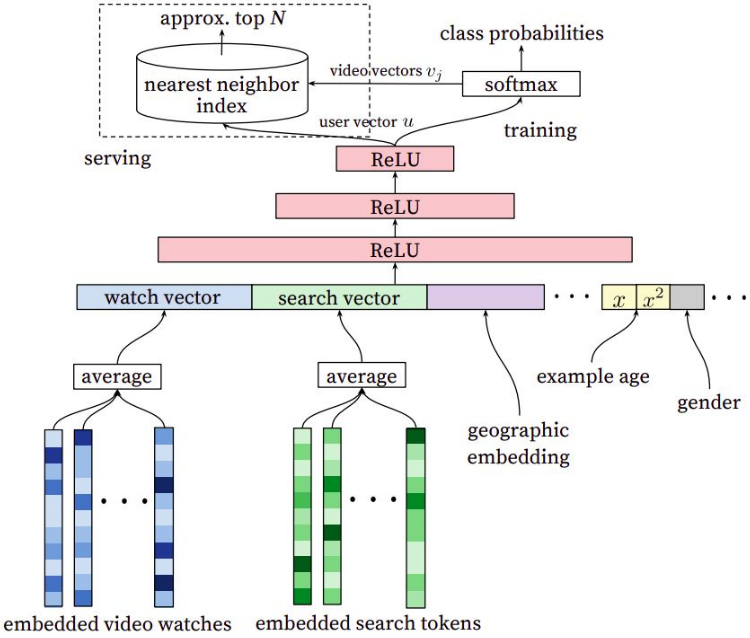


Neural Collaborative Filtering



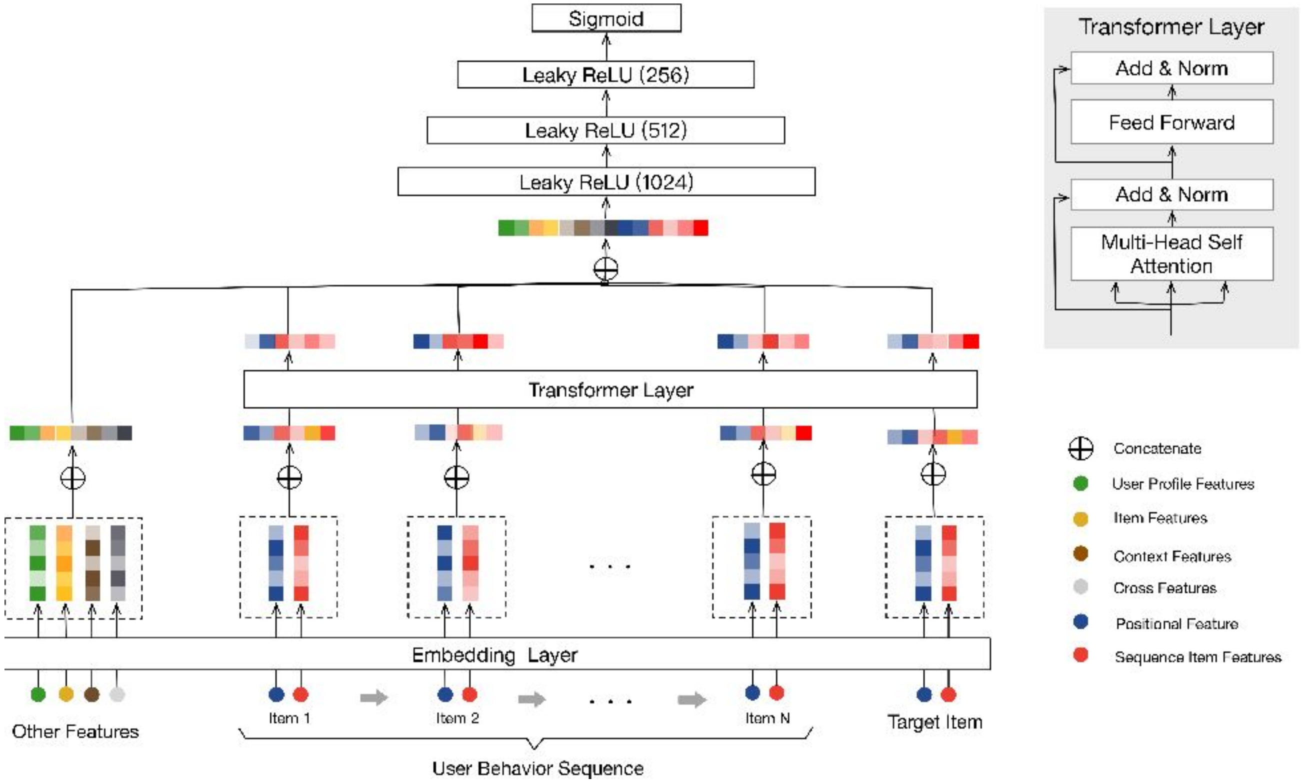
Neural collaborative filtering: Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, Tat-Seng Chua

Neural Collaborative Filtering



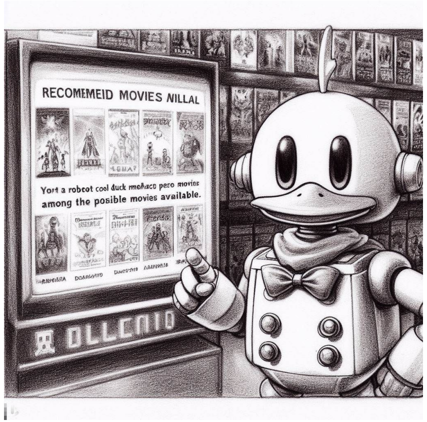
[**Deep neural networks for youtube recommendations:** P Covington, J Adams, E Sargin]

Neural Collaborative Filtering



Behavior Sequence Transformer for E-commerce Recommendation in Alibaba Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, Wenwu Ou

Introduction to recommender systems



AI frameworks

Evaluating recommender systems

Offline Evaluation

Evaluating the model before deployment:

- Using history of user-item interactions
- Choose a train/test split strategy: random-split, time-based split
- Filter users or items without enough interactions
- Choose a metric: explicit/implicit feedbacks

Offline Evaluation Metrics

u 4 5 4 3 5

2.3 4.4 3.7 2.8 4.2

$$MAE = \frac{1}{N}\sum_{(u,i)\in\text{data}} |r_{ui} - \hat{r}_{ui}|$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{(u,i)\in\text{data}} (r_{ui} - \hat{r}_{ui})^2}$$

Explicit feedbacks:

- MAE
- RMSE

Offline Evaluation Metrics

Explicit feedbacks:

- MAE
- RMSE
- Precision
- Recall
- F1-Score

$$MAE = \frac{1}{N} \sum_{(u,i)\in \text{data}} |r_{ui} - \hat{r}_{ui}|$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{(u,i)\in \text{data}} (r_{ui} - \hat{r}_{ui})^2}$$

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

$$F1\_Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

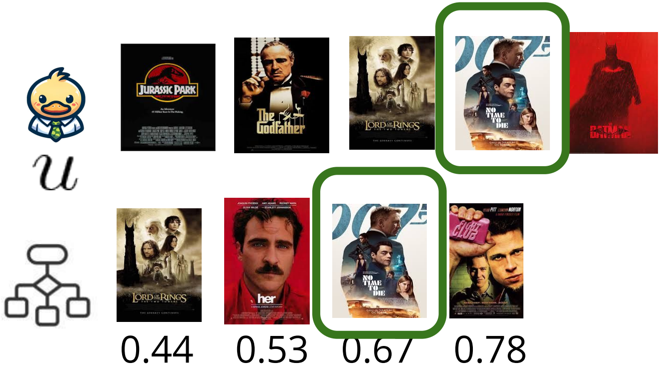
Offline Evaluation Metrics

u

0.44 0.53 0.67 0.78

Implicit feedbacks:

Offline Evaluation Metrics



Mean Reciprocal Rank (MRR)

- Find first relevant recommendation rank k: (here: k = 2)

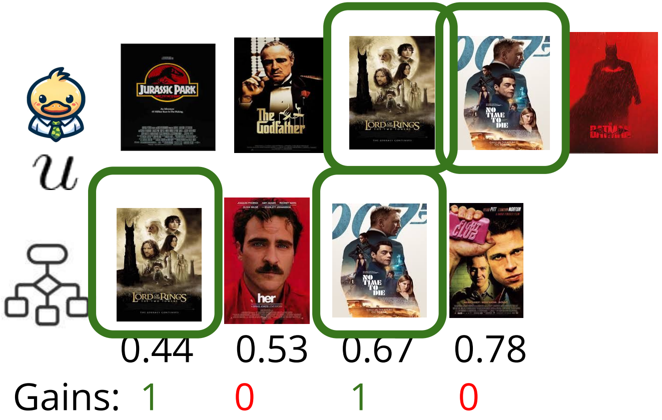
$$MRR = \frac{1}{|U|} \sum_{u \in U} \frac{1}{k_u}$$

Implicit feedbacks:

- Mean Reciprocal Rank (MRR)

Offline Evaluation Metrics

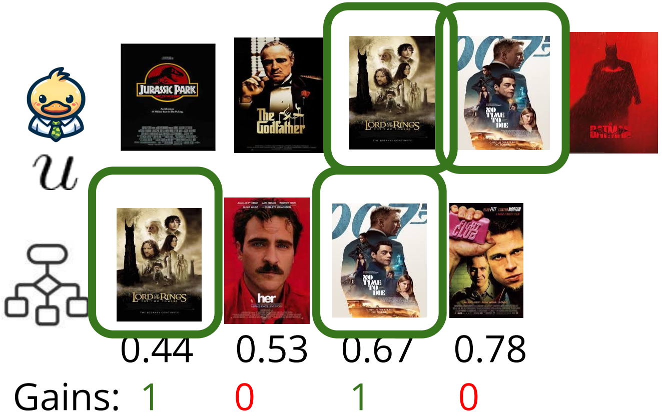
Normalized Discounted Cumulative Gain (NDCG)



Implicit feedbacks:

- Mean Reciprocal Rank (MRR)
- Normalized Discounted Cumulative Gain (NDCG)

Offline Evaluation Metrics



Normalized Discounted Cumulative Gain (NDCG)

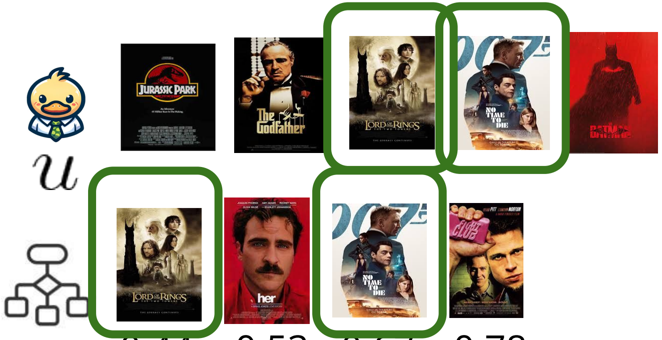
- Compute the cumulative gain:

$$DCG = \sum_{k=1}^{K} Gains = 0 + 1 + 0 + 1$$

Implicit feedbacks:

- Mean Reciprocal Rank (MRR)
- Normalized Discounted Cumulative Gain (NDCG)

Offline Evaluation Metrics



0.44 0.53 0.67 0.78

Gains: 1 0 1 0

Implicit feedbacks:

- Mean Reciprocal Rank (MRR)
- Normalized Discounted Cumulative Gain (NDCG)

Normalized Discounted Cumulative Gain (NDCG)

- Compute the cumulative gain:

$$DCG = \sum_{k=1}^{K} Gains = 0 + 1 + 0 + 1$$

- Compute discounted cumulative gain (DCG):

$$DCG = \sum_{k=1}^{K} \frac{Gains}{\log(k+1)}$$

$$= \frac{0}{\log(1+1)} + \frac{1}{\log(2+1)} + \frac{0}{\log(3+1)} + \frac{1}{\log(4+1)}$$

Offline Evaluation Metrics

0.44 0.53 0.67 0.78

Gains: 1 0 1 0

Implicit feedbacks:

- Mean Reciprocal Rank (MRR)
- Normalized Discounted Cumulative Gain (NDCG)

Normalized Discounted Cumulative Gain (NDCG)

- Compute the cumulative gain:

$$DCG = \sum_{k=1}^{K} Gains = 0 + 1 + 0 + 1$$

- Compute discounted cumulative gain (DCG):

$$DCG = \sum_{k=1}^{K} \frac{Gains}{\log(k+1)}$$

$$= \frac{0}{\log(1+1)} + \frac{1}{\log(2+1)} + \frac{0}{\log(3+1)} + \frac{1}{\log(4+1)}$$

- Compute Ideal Discounted Cumulative Gain (IDCG):

$$DCG = \frac{1}{\log(1+1)} + \frac{1}{\log(2+1)} + \frac{0}{\log(3+1)} + \frac{0}{\log(4+1)}$$

Offline Evaluation Metrics

0.44 0.53 0.67 0.78

Gains: 1 0 1 0

Implicit feedbacks:

- Mean Reciprocal Rank (MRR)
- Normalized Discounted Cumulative Gain (NDCG)

Normalized Discounted Cumulative Gain (NDCG)

- Compute the cumulative gain:

$$DCG = \sum_{k=1}^{K} Gains = 0 + 1 + 0 + 1$$

- Compute discounted cumulative gain (DCG):

$$DCG = \sum_{k=1}^{K} \frac{Gains}{\log(k+1)}$$
$$= \frac{0}{\log(1+1)} + \frac{1}{\log(2+1)} + \frac{0}{\log(3+1)} + \frac{1}{\log(4+1)}$$

- Compute Ideal Discounted Cumulative Gain (IDCG):

$$DCG = \frac{1}{\log(1+1)} + \frac{1}{\log(2+1)} + \frac{0}{\log(3+1)} + \frac{0}{\log(4+1)}$$

- Compute Normalized Discounted Cumulative Gain (NDCG):

$$NDCG = \frac{DCG}{IDCG}$$

Recommender systems in real-life

Real evaluation

Challenges of offline Evaluation:

- Biais in user interactions and ratings
- Highly sparse data

Real evaluation

Challenges of offline Evaluation:

- Biais in user interactions and ratings
- Highly sparse data

Online evaluation:

- Click-through rates
- Adoption and conversion (percentage of song listened, percentage of products bought, …
- Global revenue
- User behaviour and engagement (are the user coming more often? Do they stay longer?)
- A/B testing

Tricks

- Be careful to seasonality and behavioral change
- Be careful to bots
- Choose recommendation strategy depending on context
- Recommendation position is important
- Explaining why an item was recommended improves conversion rate
- Best sellers are an efficient option

Introduction to recommender systems

AI frameworks