AWS re:Invent

MDS303-R

Evolution of personalization/recommendation for video workflows

Liam Morrison
Principal Solutions Architect
Amazon Web Services

Shaun McCarthy
Senior Manager, Software Dev
Amazon Prime Video

AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

Related breakouts

AIM304 - Build a content-recommendation engine with Amazon Personalize

AIM323 - Delight your customers with ML-based personalized recommendations

ADM402 - Testing techniques for measuring personalization

AIM424 - Accelerate experimentation with personalization models

STP304 - Predicting what your customers really, really want

RET310 - Create higher-quality recommendations in your e-commerce platform

Agenda

Overview of recommendation engines

Common approaches to recommendation engines

Amazon Prime Video’s evolution

Overview of recommendation engines

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What is a recommendation engine?

A **recommender system** or a **recommendation engine** is a subclass of information filtering systems that seeks to predict the "rating" or "preference" a user would give to an item

Wikipedia

Non-personalized recommendations

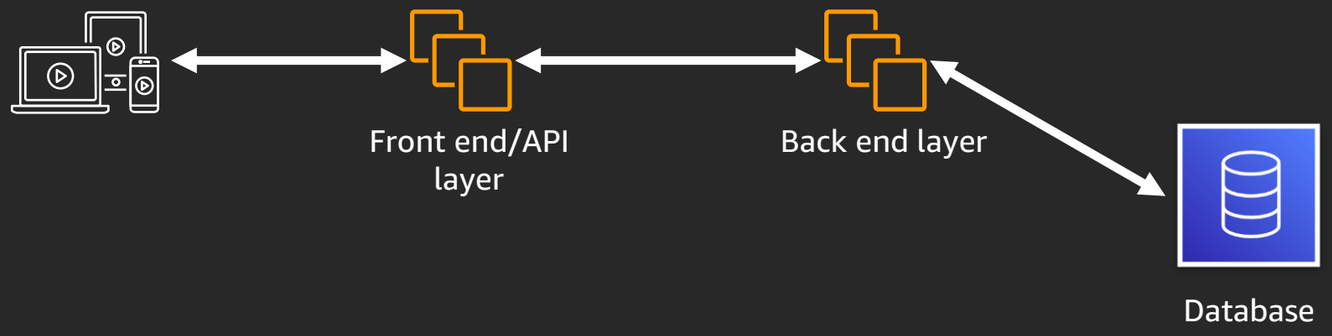
- Curated shelf at store
- Zagat survey
- Initial home page of streaming site (when not logged in/without user data)
- AFI's 100 Years...100 Movies
- Website/newspaper best of lists

What is personalization?

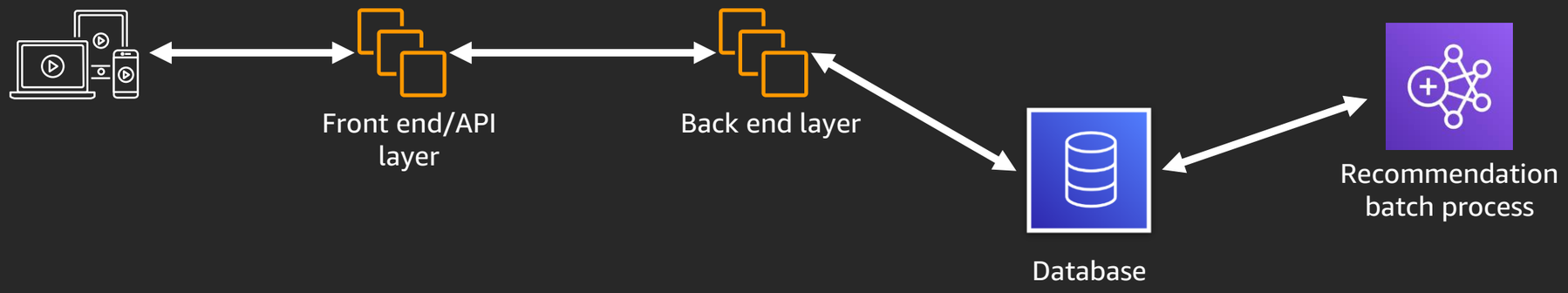
Personalization (broadly known as customization) consists of tailoring a service or a product to accommodate specific individuals, sometimes tied to groups or segments of individuals

Wikipedia

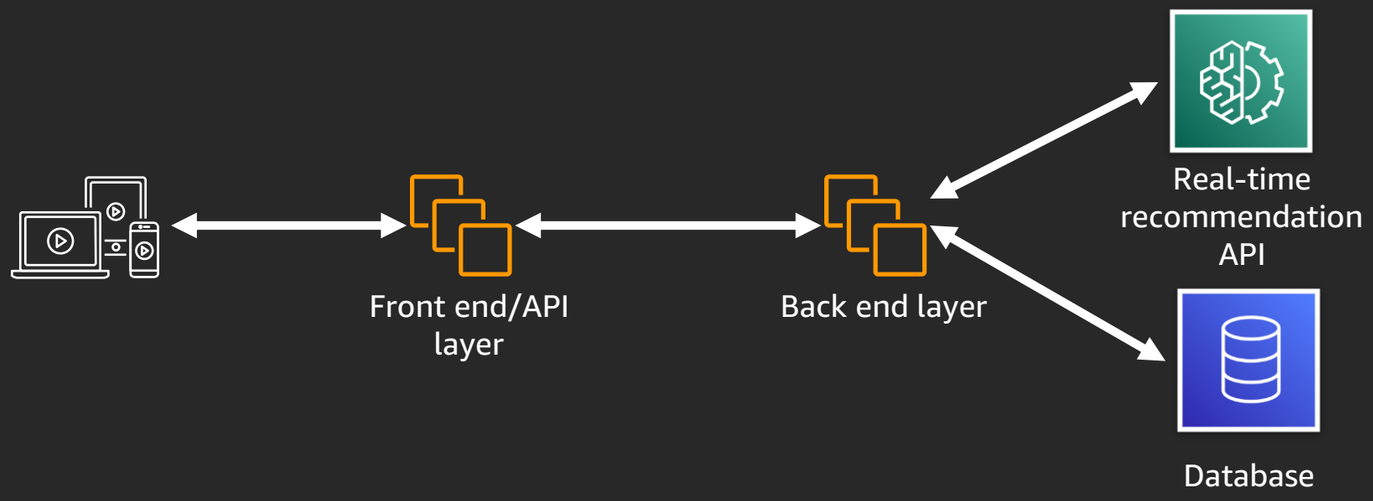
Simplified recommendation architecture



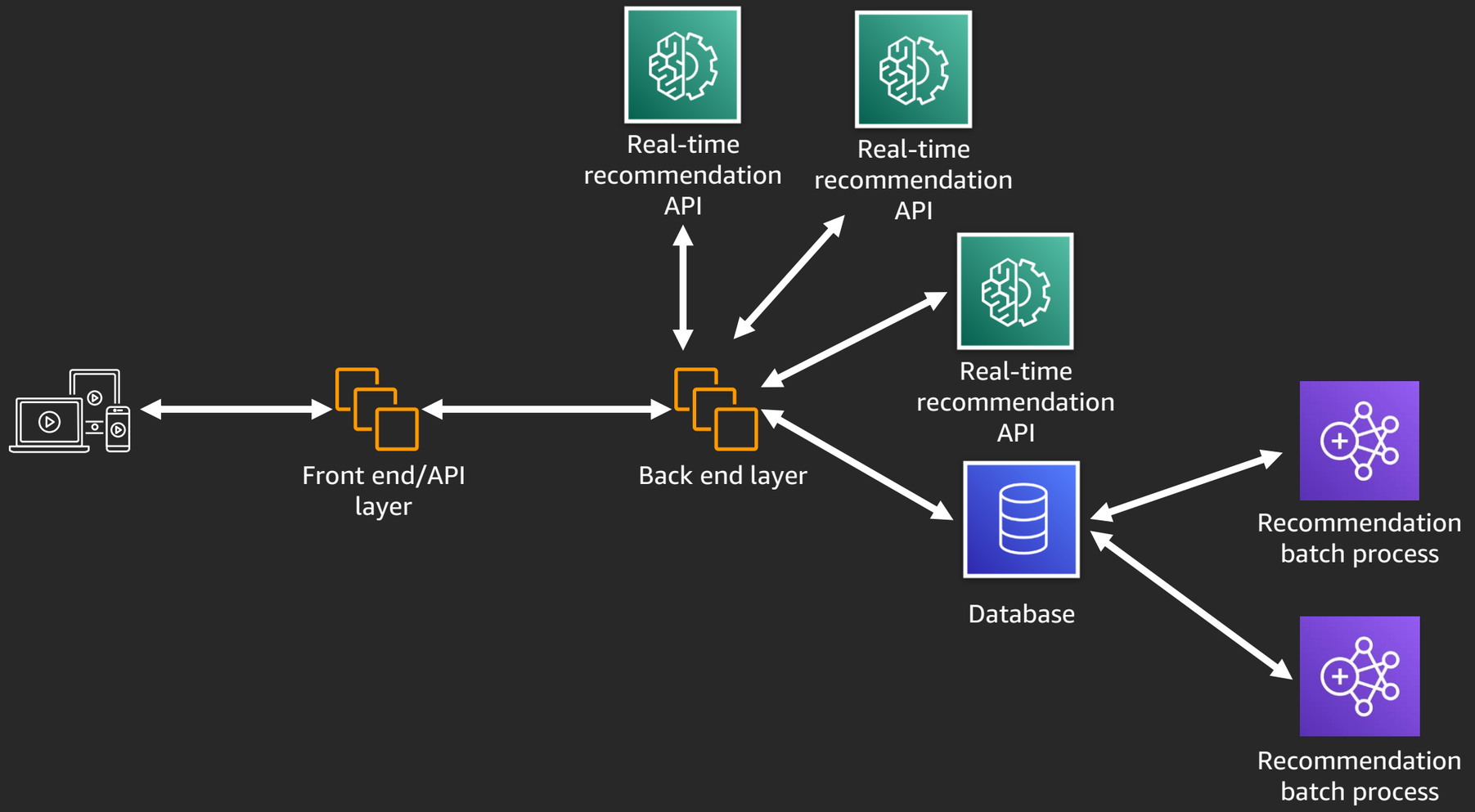
Simplified recommendation architecture



Simplified recommendation architecture



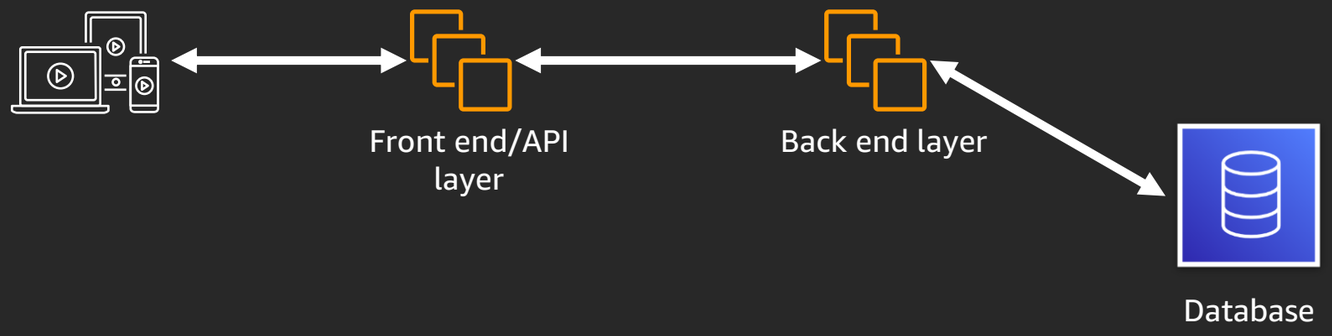
~~Simplified~~ recommendation architecture



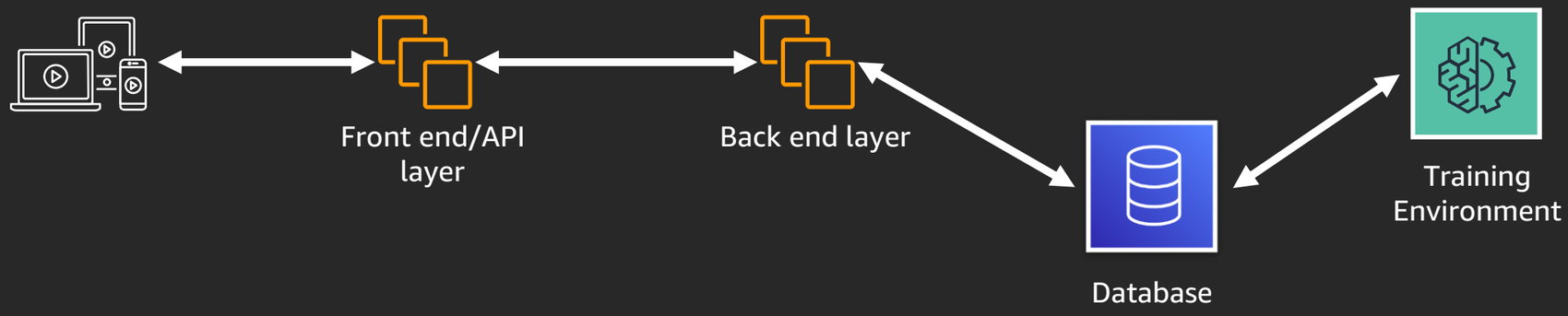
Types of recommendation/personalization in a video workflow

- Homepage video recommendations (billboard/featured)
- Genre/category (top x movies, comedies, dramas for each user)
- Item-to-item similarities (if you like this, then you will like x)
- Personal re-ranking (search, genre)
- Layout customization (order of carousel, etc.)
- Artwork/thumbnail customization (sci-fi aspects vs. horror aspects)
- Notifications (email, push, etc.)

Simplified recommendation architecture



Simplified recommendation architecture



Explicit feedback

Users state their interest in a piece of content

- Numerical rating (x/100)
- Star ratings ()
- Thumbs up / Upvote (👍 👎)
- Like
- Survey

Pros

- Users preferences are stated explicitly

Cons

- Very few people actually rate/vote
- Many people only rate on extremes (love or hate)
- Preferences can change
- Hard to calibrate

Explicit feedback: MovieLens 20M

MovieLens 20m describes 5-star rating and free-text tagging activity from MovieLens, a movie recommendation service. It contains 20,000,263 ratings and 465,564 tag applications across 27,278 movies. These data were created by 138,493 users between January 09, 1995 and March 31, 2015.

27k movies * 138k users = 3.7B possible ratings

20M actual ratings

99.5% of ratings are not identified

- Viewers don’t watch every movie
- Viewers do not rate every movie they watch

https://grouplens.org/datasets/movielens/20m/

Implicit feedback

Feedback is derived based on the actions taken by a user

- Click/watch content
- Purchasing
- Subscribe/follow
- Exiting content before it’s complete
- Add to cart

Pros

- Expressing implicit feedback is mandatory to view content (all the data)
- Harder to game

Cons

- Does not capture if user actually liked the content, just that they clicked: no direct negative feedback
- Noise

AWS services for recommendation engines

- Amazon Personalize
- Amazon SageMaker (Factorization machines, Gluon, K-NN, BYO)
- Amazon EMR (Spark, Apache MXNet on AWS, TensorFlow)
- Amazon Elastic Compute Cloud (Amazon EC2)/AWS Deep Learning AMI/Container
- Partner solutions

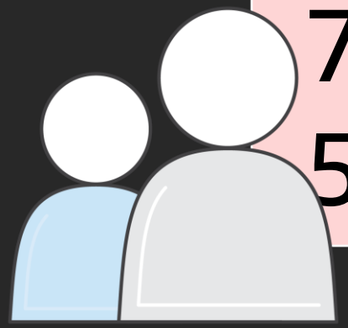
Common approaches to recommendation engines

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Content-based filtering

Generate recommendations based on **known** user preferences provided either explicitly or implicitly

UserId	Rock	Jazz	HipHop	Classical
76 53	5	2	3	1





```
{
  "itemId": 842,
  "rock": 0.78,
  "jazz": 0.05,
  "hiphop": 0.20,
  "classical": 0.01
}
```

Content-based filtering

Advantages

- Little/no “cold start” for new items
- Not dependent on large user base
- Easier to implement (compared to collaborative filtering)
- Interpretable recommendations

Disadvantages

- Lack of diversity and serendipity
- Doesn’t capture interdependencies or complex behaviors
- Cold start for new users

Collaborative filtering

- Based on **item-to-item** or **user-user** relationships
- Derived from **explicit** and **implicit** features
- Recommendations are based on **other user's experiences**



Industry Report

Amazon.com Recommendations

Item-to-Item Collaborative Filtering

Greg Linden, Brent Smith, and Jeremy York • *Amazon.com*

Recommendation algorithms are best known for their use on e-commerce Web sites,[1] where they use input about a customer's interests to generate a list of recommended items. Many applications use only the items that customers purchase and explicitly rate to represent their interests, but they can also use other attributes, including items viewed, demographic data, subject interests, and favorite artists.

At Amazon.com, we use recommendation algorithms to personalize the online store for each customer. The store radically changes based on customer interests, showing programming titles to a software engineer and baby toys to a new mother. The click-through and conversion rates — two important measures of Web-based and email advertising effectiveness — vastly exceed those of untargeted content such as banner advertisements and top-seller lists.

There are three common approaches to solving the recommendation problem: traditional collaborative filtering, cluster models, and search-based methods. Here, we compare these methods with our algorithm, which we call *item-to-item collaborative filtering*. Unlike traditional collaborative filtering, our algorithm's online computation scales independently of the number of customers and number of items in the product catalog. Our algorithm produces recommendations in realtime, scales to massive data sets, and generates high-quality recommendations.

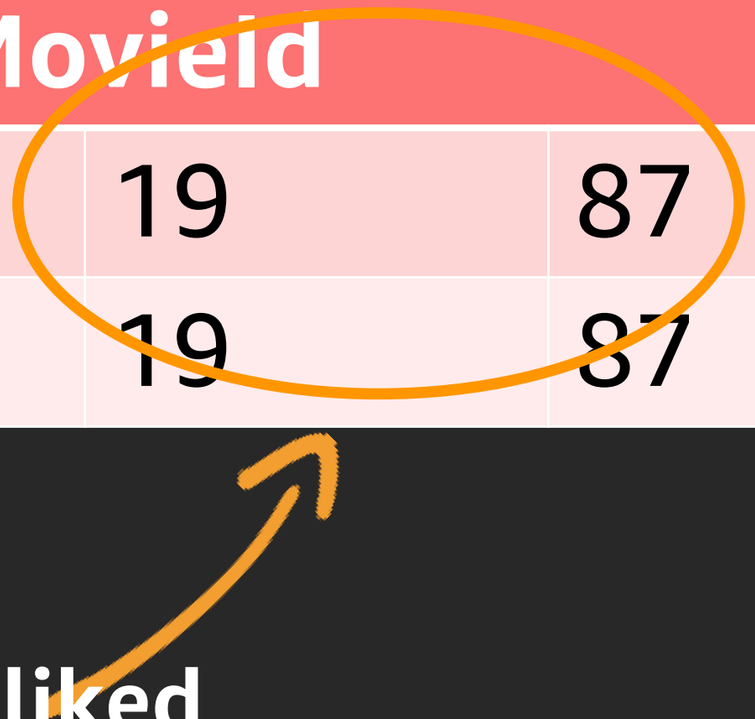
Recommendation Algorithms

Most recommendation algorithms start by finding a set of customers whose purchased and rated items overlap the user's purchased and rated items.[2] The algorithm aggregates items from these similar customers, eliminates items the user has

https://www.cs.umd.edu/~samir/498/Amazon-Recommendations.pdf

Collaborative filtering example

UserId	LikedMovieId			
123	12	19	87	171
456	15	19	87	231



Movies 19, 87 are liked by multiple users

Collaborative filtering example

UserId	Liked MovieId			
123	12	19	87	171
456	15	19	87	231

Likely recommendation

Collaborative filtering

- Advantages

- Diversity in recommendations
- Serendipity and discoverability
- Leverages inter-user or inter-item information
- Better predictions
- Scalable
- Feature extraction is easier

- Disadvantages

- Cold start problem
- User variation in rating
- Lower interpretability compared to content-based
- Complicated to implement compared to content-based
- Computationally expensive

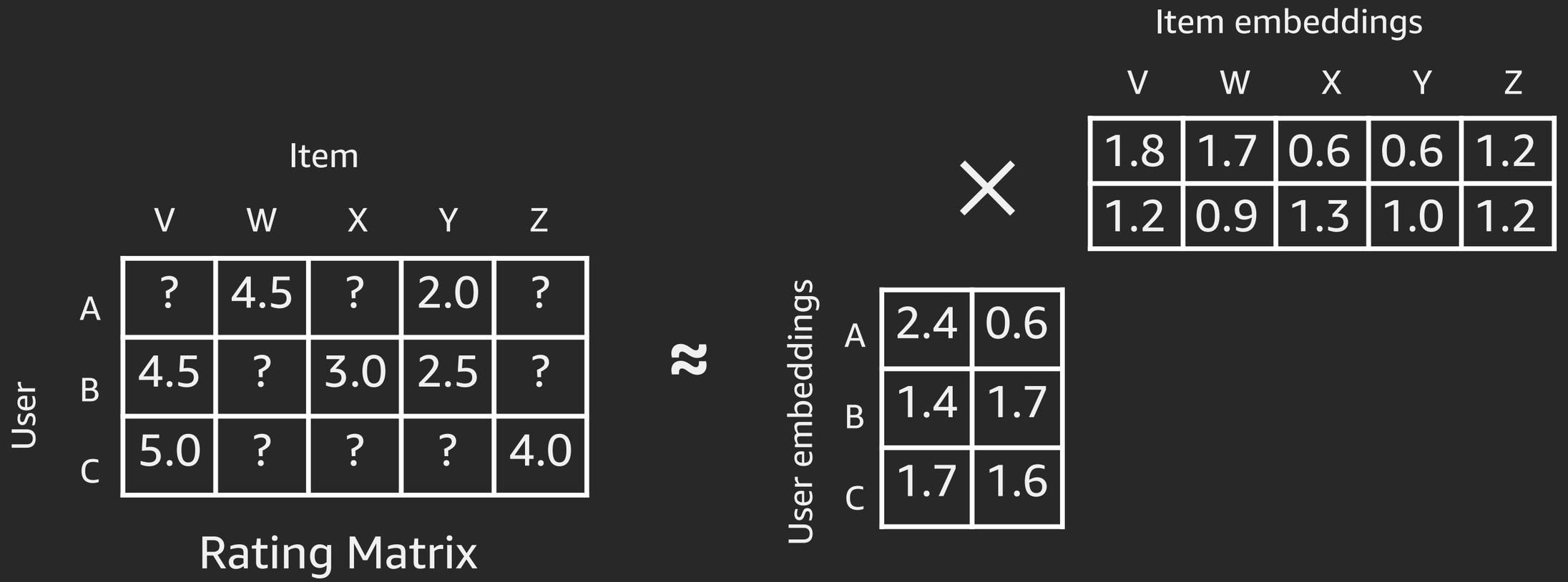
Matrix factorization

User \ Item	V	W	X	Y	Z
A	?	4.5	?	2.0	?
B	4.5	?	3.0	2.5	?
C	5.0	?	?	?	4.0

Rating Matrix

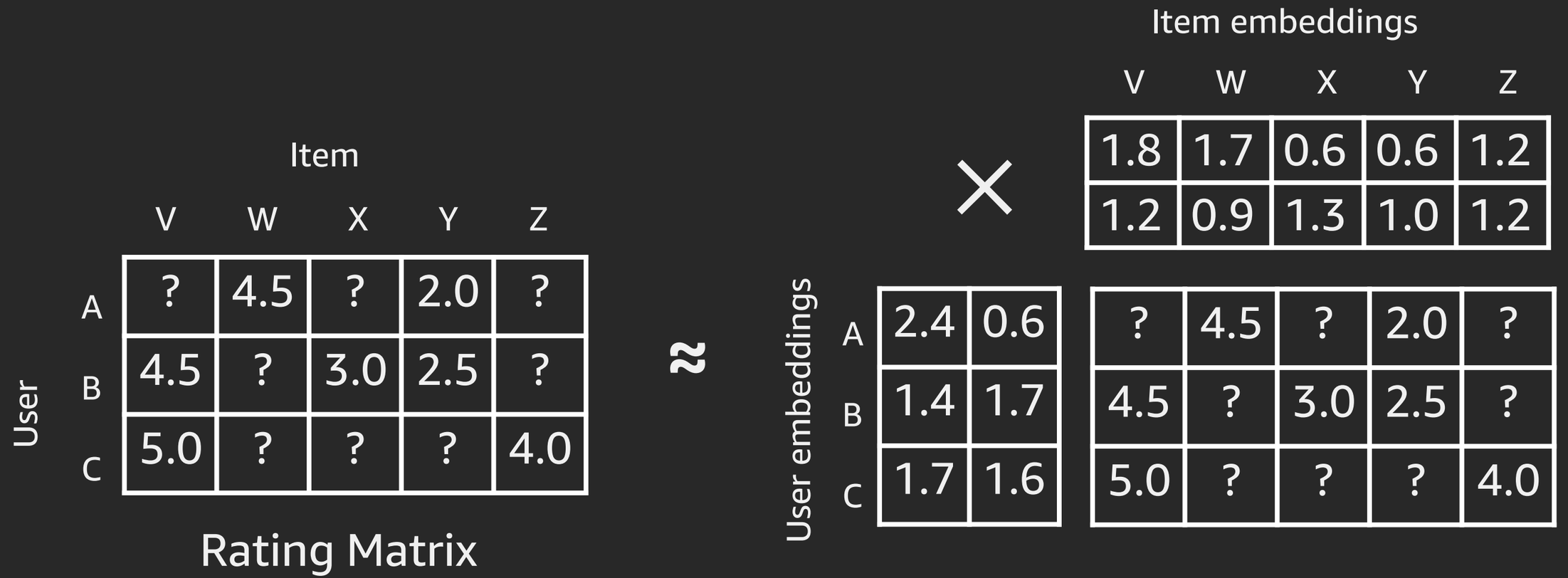
https://www.csie.ntu.edu.tw/~b97053/paper/Rendle2010FM.pdf

Matrix factorization



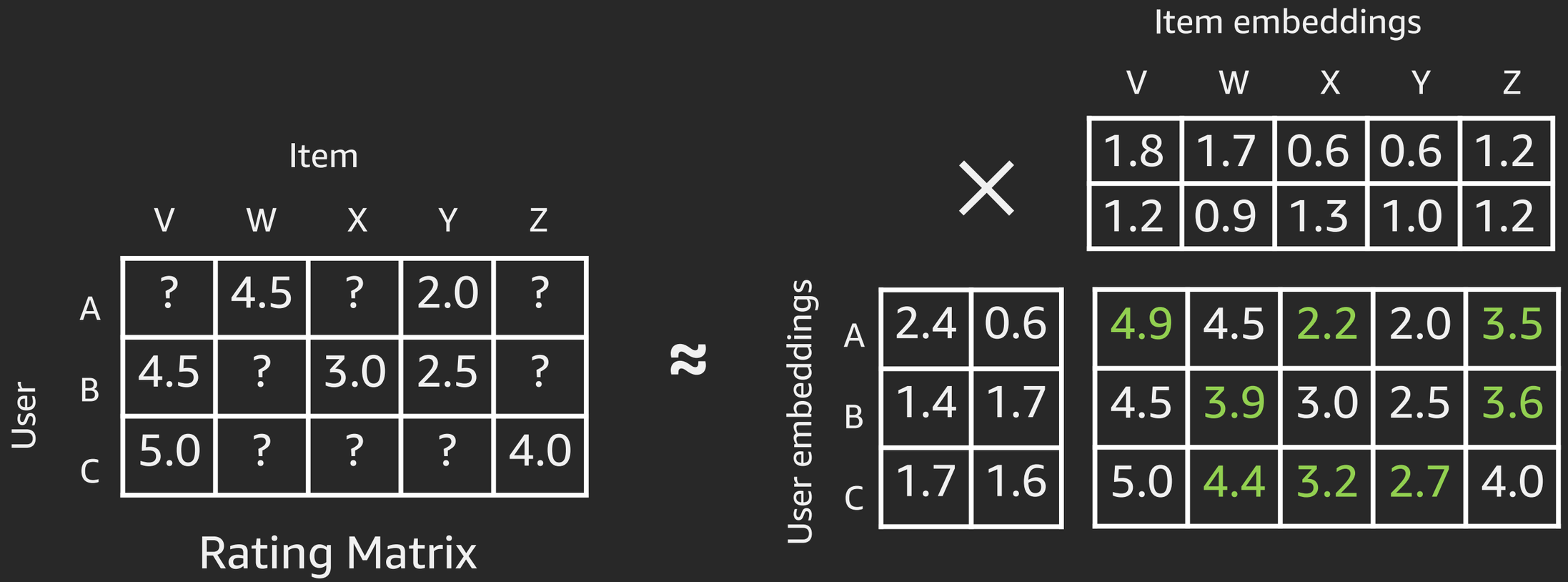
https://www.csie.ntu.edu.tw/~b97053/paper/Rendle2010FM.pdf

Matrix factorization



https://www.csie.ntu.edu.tw/~b97053/paper/Rendle2010FM.pdf

Matrix factorization

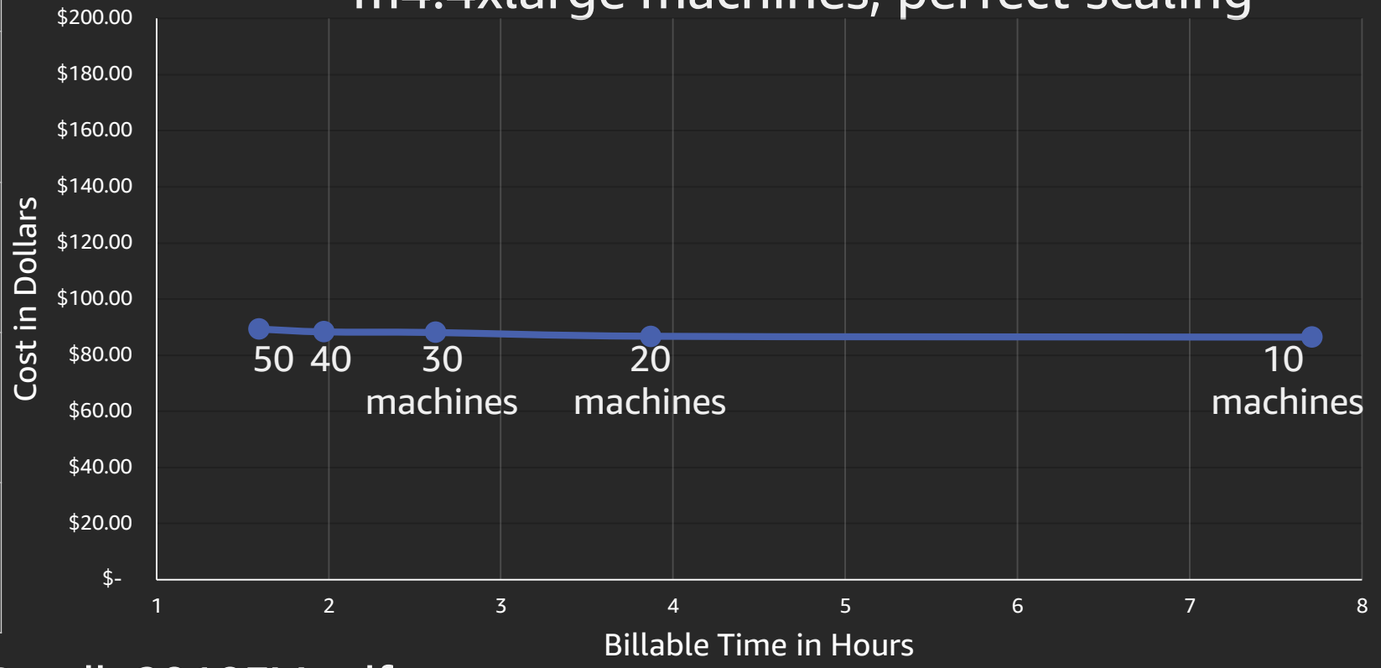


Factorization machines in Amazon SageMaker

$$\tilde{y} = w_0 + \langle w_1, x \rangle + \sum_{i,j>i} x_i x_j \cdot \langle v_i, v_j \rangle$$

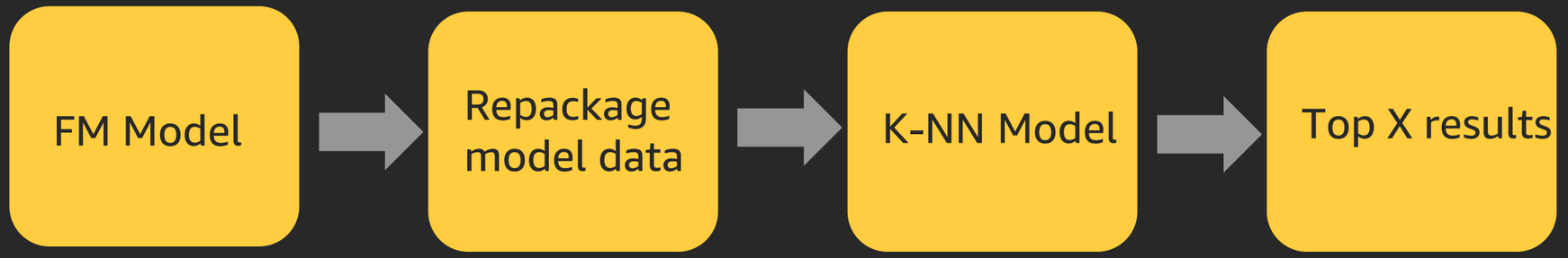
	Log loss	F1 Score	**Seconds**
Amazon SageMaker	0.494	0.277	820
Other (10 Iter)	0.516	0.190	650
Other (20 Iter)	0.507	0.254	1300
Other (50 Iter)	0.481	0.313	3250

Click Prediction 1 TB advertising dataset, m4.4xlarge machines, perfect scaling



https://www.csie.ntu.edu.tw/~b97053/paper/Rendle2010FM.pdf

Extending factorization machines with K-NN




```
Recommended movie Ids for user #812 : [760, 220, 252, 755, 542, 984, 808, 1230, 369, 802, 43, 5, 781, 894, 2, 1033, 783, 940,
1089, 978, 554, 476, 368, 829, 795, 391, 145, 229, 159, 1139, 441, 259, 871, 1419, 264, 944, 682, 300, 934, 125, 815, 375, 819,
358, 1037, 878, 63, 977, 1036, 384, 540, 561, 563, 294, 546, 769, 121, 866, 595, 450, 717, 120, 281, 21, 452, 328, 827, 370, 720,
986, 225, 456, 1013, 826, 383, 873, 928, 687, 118, 472, 890, 148, 105, 1047, 254, 1090, 748, 877, 332, 29, 756, 243, 926, 930,
597, 322, 1028, 38, 678, 323]

Movie distances for user #812 : [1.0014, 1.0016, 1.0019, 1.0083, 1.0119, 1.0125, 1.0146, 1.0148, 1.0181, 1.0211, 1.0217, 1.0219,
1.0238, 1.0253, 1.0293, 1.0309, 1.0316, 1.0324, 1.0347, 1.0365, 1.0391, 1.0432, 1.0445, 1.0471, 1.0487, 1.0493, 1.0528, 1.0612,
1.0703, 1.0802, 1.089, 1.0907, 1.0914, 1.0952, 1.0959, 1.0974, 1.0987, 1.0993, 1.105, 1.1067, 1.1087, 1.1108, 1.1114, 1.1181,
1.1228, 1.1233, 1.1239, 1.1253, 1.1318, 1.1345, 1.1432, 1.1459, 1.1466, 1.1475, 1.1481, 1.1489, 1.1521, 1.1691, 1.1745, 1.1931,
1.2114, 1.225, 1.2255, 1.2322, 1.2389, 1.2572, 1.2587, 1.2667, 1.2722, 1.2844, 1.2912, 1.2973, 1.3029, 1.3064, 1.3167, 1.3443,
1.3444, 1.3533, 1.365, 1.3697, 1.3906, 1.3982, 1.4212, 1.4244, 1.4342, 1.4553, 1.4676, 1.468, 1.469, 1.5009, 1.5084, 1.5096,
1.5176, 1.5227, 1.5267, 1.5927, 1.6084, 1.8161, 1.9924, 2.0966]
```

https://aws.amazon.com/blogs/machine-learning/extending-amazon-sagemaker-factorization-machines-algorithm-to-predict-top-x-recommendations/

Demo

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deep learning recommendation engines

- Auto-encoders
- Restricted boltzmann machines (RBM)
- Recurrent neural networks (RNN)
- Convolutional neural networks (CNN)
- Natural Language Processing (NLP)
- Deep reinforcement learning
- Adversarial networks
- Hybrid models

Deep learning recommendation engines

Advantages

- Capable of modeling complex interactions (non-linear)
- Can model sequential data (user behavior)
- Flexible
- Less need to focus on feature extraction

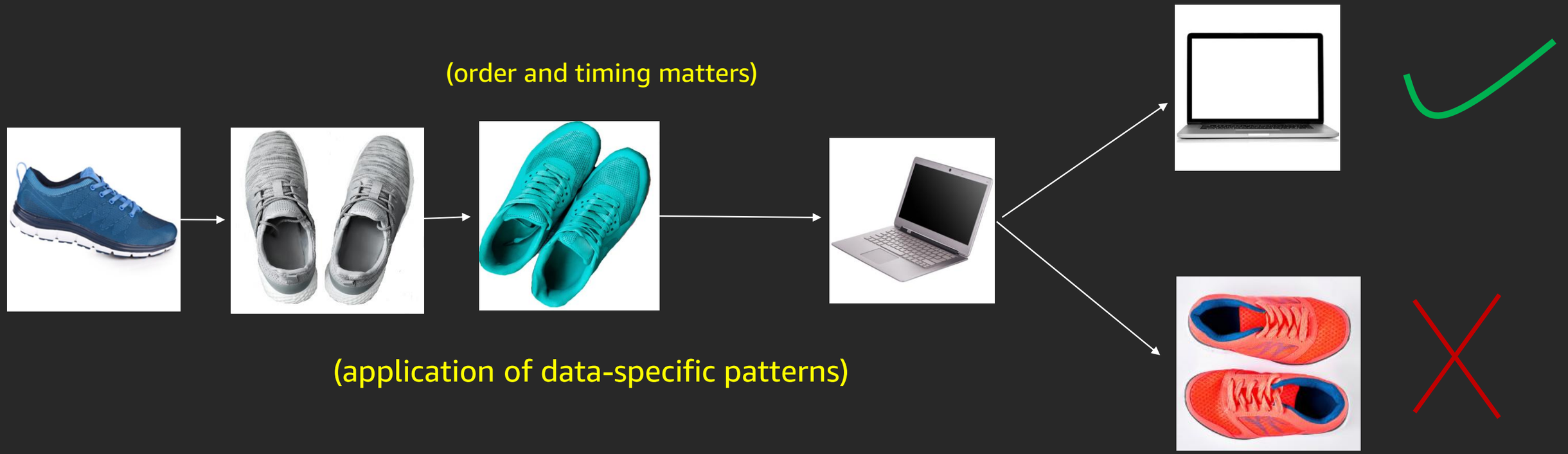
Disadvantages

- Requires a lot of data
- Very low interpretability
- Complicated to implement
- Computationally expensive

Modeling for personalization

Fundamental insight

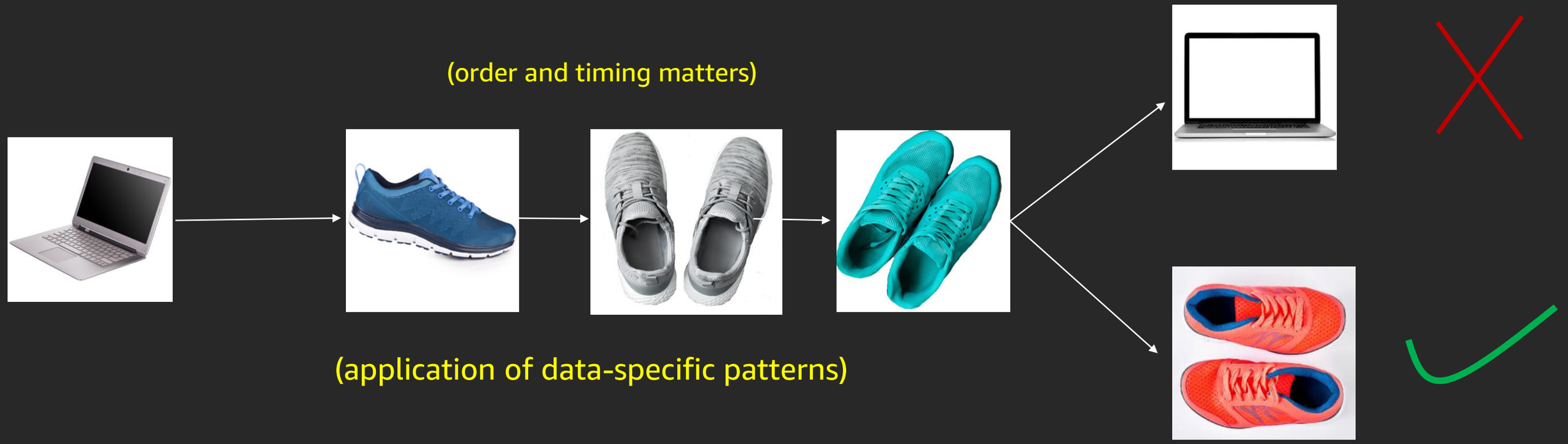
The evolution of historical interest and dis-interest is a good indicator of future preferences



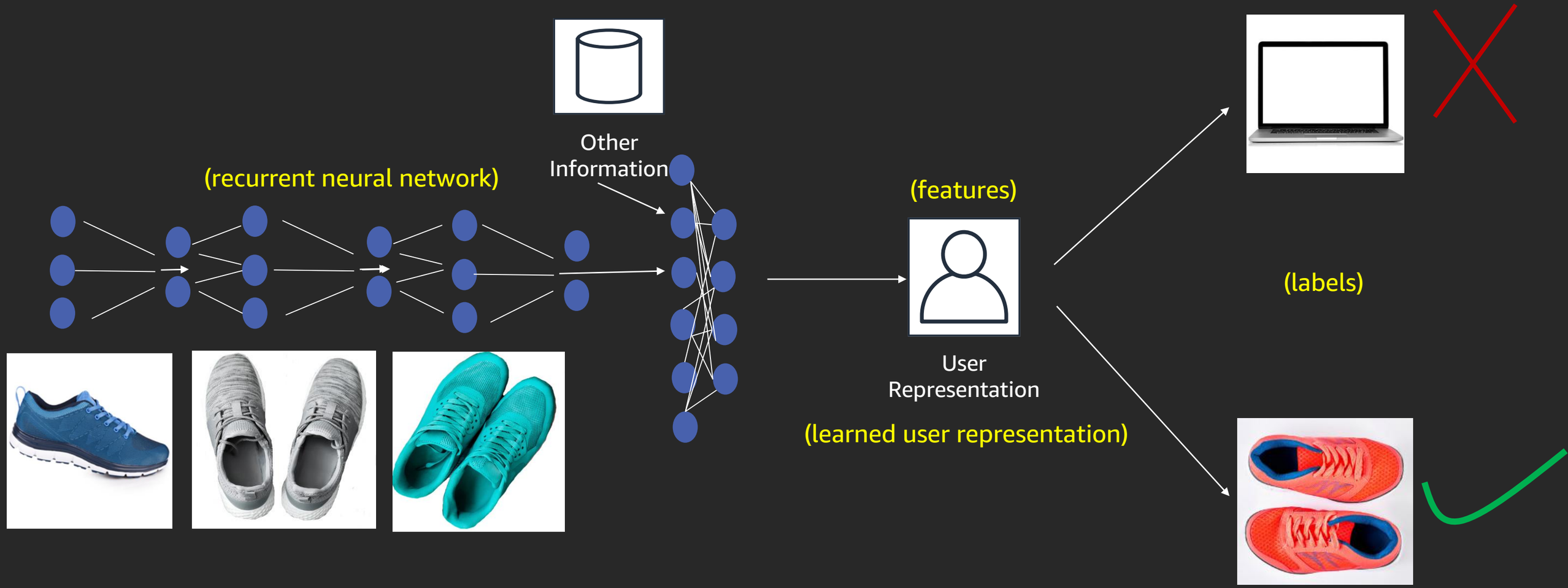
Modeling for personalization

Fundamental insight

The evolution of historical interest and dis-interest is a good indicator of future preferences

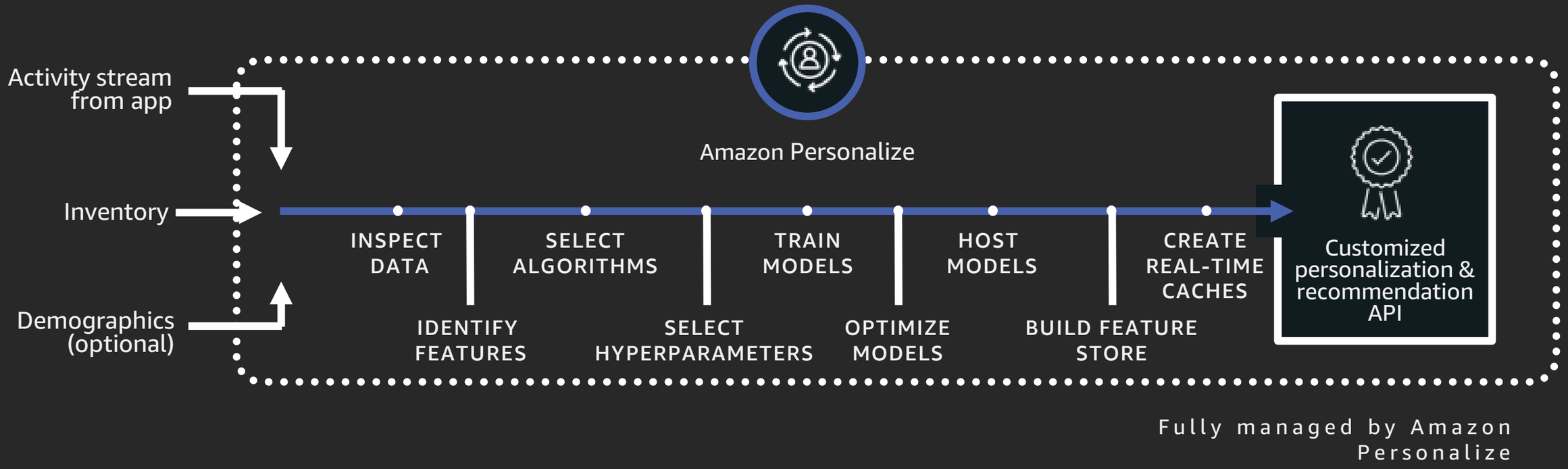


Sequential modeling

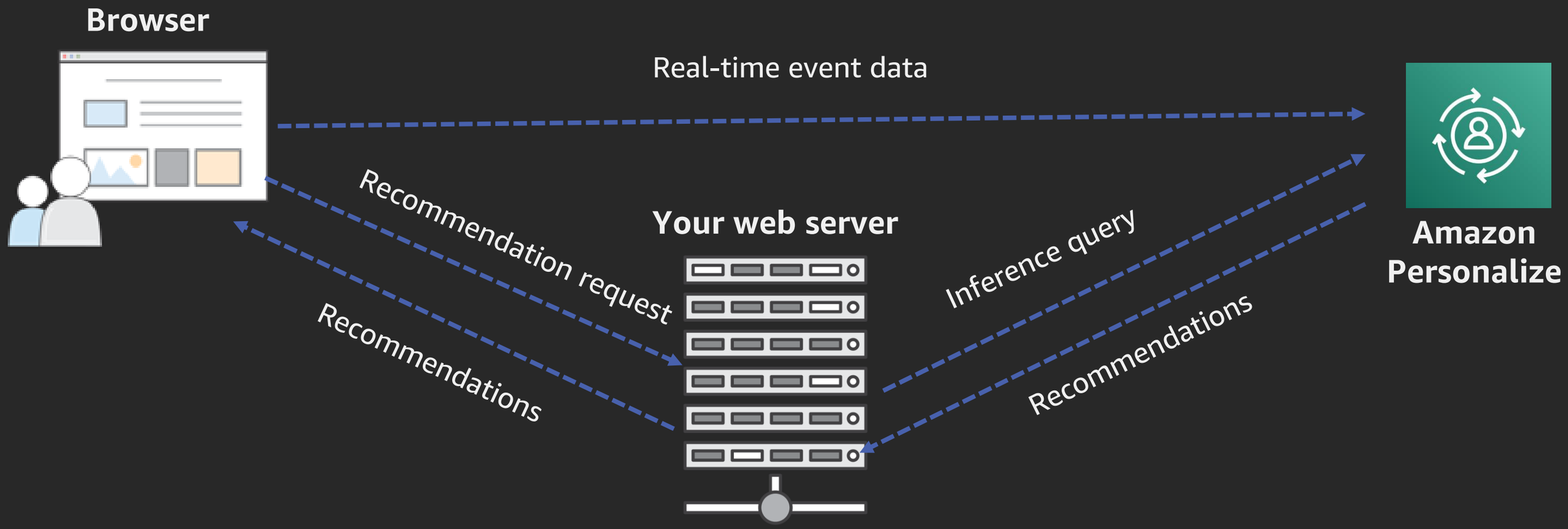


https://openreview.net/pdf?id=ByzxsrrkJ4

Amazon Personalize: Machine learning personalization and recommendations



Real-time personalization for web applications



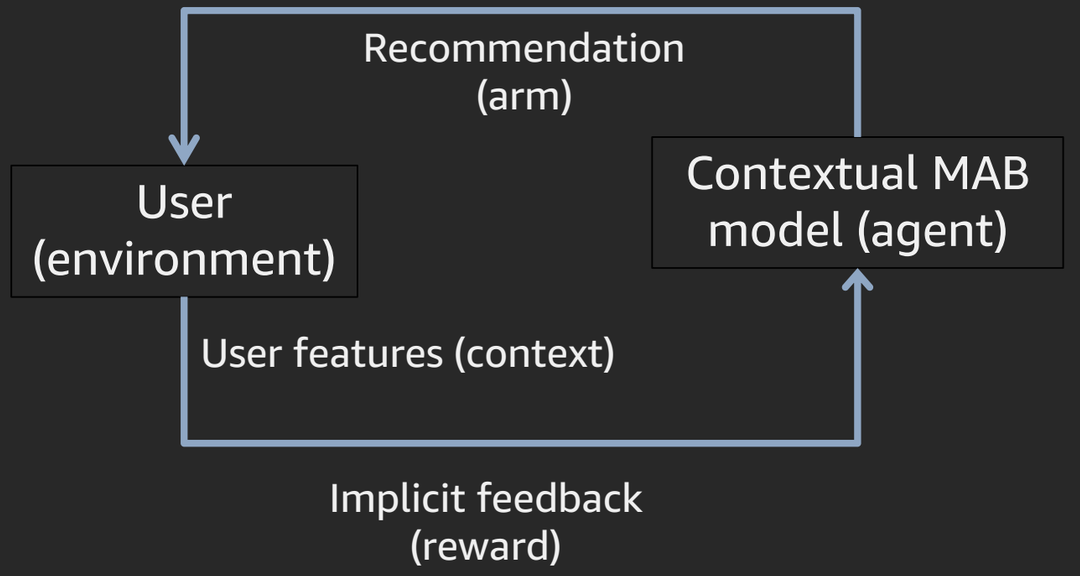
Demo

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Multi-armed bandit

Multi-armed bandit is a probability theory problem that comes from imagining a gambler at a row of slot machines (sometimes known as a "one-armed bandit"), who has to decide which machines to play, how many times to play each machine, in which order to play them, and whether to continue with the current machine or try a different machine





Amazon SageMaker RL contextual bandits

https://aws.amazon.com/blogs/machine-learning/power-contextual-bandits-using-continual-learning-with-amazon-sagemaker-rl/

Prime Video recommendations

Shaun McCarthy
Sr. Manager, Prime Video Personalization

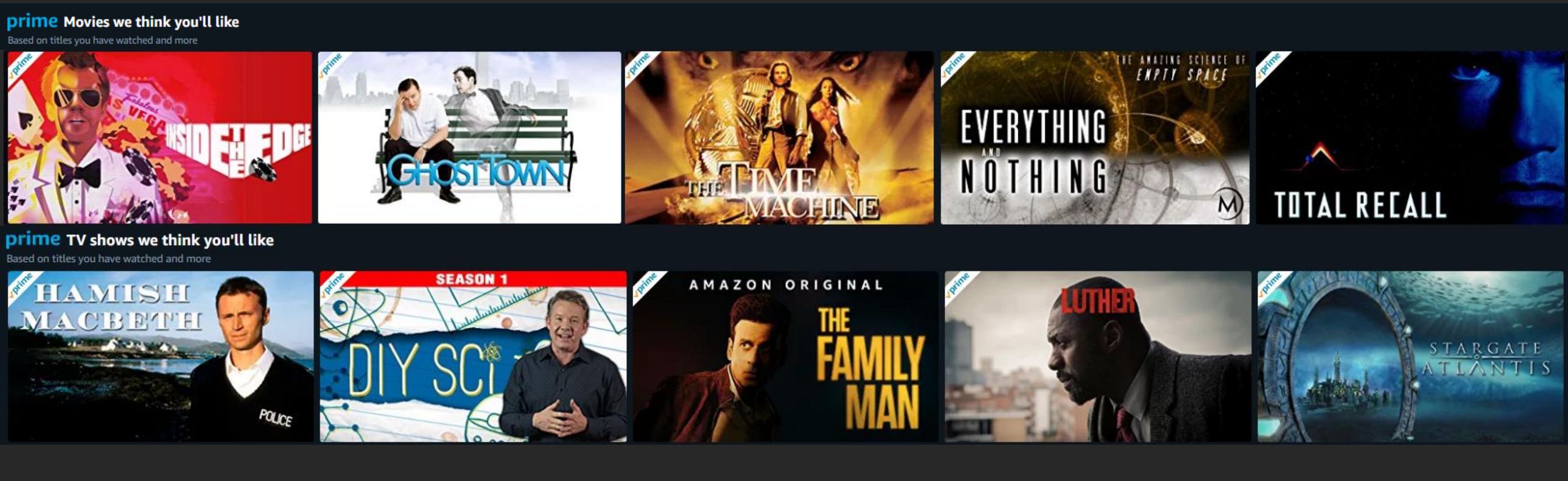
AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

Case study—Recommended For You

- Describe the customer problem
- The evolution of our solutions
- Learnings along the way



Defining the customer problem

- How would you optimize the layout?
- Start with the customer
  - Spearfishing
  - Some idea
  - No idea



Physical store

- Typical solution
  - **Spearfishing:** Recent & popular movies up front; video clerk
  - **Some idea:** Sections for genres
  - **No idea**: Browse or video clerk
- Shaped by physical constraints
  - Same layout for all customers
  - Expensive to move titles around
  - Don’t know anything about the customer
  - Shelf space

Moving online

- What's different?
  - Changing the layout online is cheap
  - Store can be different for each customer
  - We know more about the customer
  - We can show everything
- Solutions depend on the mission
  - **Spearfishing:** Search
  - **Some idea:** Carousels instead of sections
  - **No idea:** Personalized recommendations

Recommendations at scale

- 100 million+ Prime customers
- 200+ countries & territories worldwide
- Extensive catalog
  - Prime / Originals
  - HBO / Showtime / Acorn / etc.
  - Rent / purchase recent releases
  - Rent / purchase back catalog

Measuring success



- Maximize long-term engagement
  - We want to delight customers again and again
  - *A/B Testing*
- Did we get it right?
  - Recommend what people will enjoy
  - *Precision (suggested & watched / suggested)*


AMAZON ORIGINAL
THE
big sick


AMAZON ORIGINAL
The STINKY & DIRTY Show!


AMAZON ORIGINAL
CREATIVE
CONTROL


AMAZON ORIGINAL
DON'T WORRY,
HE WON'T GET FAR ON FOOT


AMAZON ORIGINAL
generation
wealth


AMAZON ORIGINAL
BRITTANY RUNS
A MARATHON


AMAZON ORIGINAL
ABSENTIA


AMAZON ORIGINAL
THE
BOYS


PRIME ORIGINAL
CATASTROPHE


AMAZON ORIGINAL
FLEABAG
EMMY WINNER


PRIME ORIGINAL
HANNA


PRIME ORIGINAL
HOMECOMING


AMAZON ORIGINAL
Creative
galaxy


AMAZON ORIGINAL
TOM CLANCY'S
JACK
RYAN
NEW SEASON


AMAZON ORIGINAL
LORE


AMAZON ORIGINAL
Wishenpoof


AMAZON ORIGINAL
LORENA


AMAZON ORIGINAL
Modern Love
NEW SERIES


AMAZON ORIGINAL
THE MAN IN THE
HIGH CASTLE
FINAL SEASON


AMAZON ORIGINAL
Tumble
Leaf


AMAZON ORIGINAL
ALL OR
NOTHING
AN NFL FILMS PRODUCTION


AMAZON ORIGINAL
ONE CHILD NATION


AMAZON ORIGINAL
sneaky pete


PRIME ORIGINAL
THE
TICK


PRIME ORIGINAL
KATE BECKINSALE
The WIDOW

Getting started

- Options
  - Content-based filtering
  - Collaborative filtering
  - Hybrid solutions
- Keep it simple
  - Quick to deliver value
  - Early data on where to improve
  - Easy to debug


AMAZON ORIGINAL
THE
big sick


AMAZON ORIGINAL
The STINKY & DIRTY Show!


AMAZON ORIGINAL
CREATIVE
CONTROL


AMAZON ORIGINAL
DON'T WORRY,
HE WON'T GET FAR ON FOOT


AMAZON ORIGINAL
generation
wealth


AMAZON ORIGINAL
BRITTANY RUNS
A MARATHON


AMAZON ORIGINAL
ABSENTIA


AMAZON ORIGINAL
THE
BOYS


PRIME ORIGINAL
CATASTROPHE


AMAZON ORIGINAL
FLEABAG
EMMY WINNER


PRIME ORIGINAL
HANNA


PRIME ORIGINAL
HOMECOMING


AMAZON ORIGINAL
Creative
galaxy


AMAZON ORIGINAL
TOM CLANCY'S
JACK
RYAN
NEW SEASON


AMAZON ORIGINAL
LORE


AMAZON ORIGINAL
Wishenpoof


AMAZON ORIGINAL
LORENA


AMAZON ORIGINAL
Modern Love
NEW SERIES


AMAZON ORIGINAL
THE MAN IN THE
HIGH CASTLE
FINAL SEASON


AMAZON ORIGINAL
Tumble
Leaf


AMAZON ORIGINAL
ALL OR
NOTHING
AN NFL FILMS PRODUCTION


AMAZON ORIGINAL
ONE CHILD NATION


AMAZON ORIGINAL
sneaky pete


PRIME ORIGINAL
THE
TICK


PRIME ORIGINAL
KATE BECKINSALE
The WIDOW

Comedy











Action











Kids









Recs









Content-based filtering: Results

- What worked?
  - Easy to debug
  - Recommendations were somewhat appropriate
  - Customer engagement increased
- What did not?
  - Recommended the same titles to large groups of people
  - Felt “static”
  - Customers aren’t that simple
  - Low quality of long-tail metadata
    - Global and new offer types just around the corner

Measuring success: Updated

- Maximize long-term engagement
  - We want customers to come back, again and again
- Did we get it right?
  - Recommend what people will enjoy
- Comprehensive
  - Surface titles deep from within the catalog
  - *Converted coverage (uniquely recommended and watched)*

Collaborative filtering

Customer A:



Customer B:



Customer C:



Collaborative filtering: Results

- What worked?
  - Recommendations were more appropriate (~2x)
  - Recommendations were more comprehensive
  - Customer engagement increased
- What did not?
  - Recommended the same titles again and again to same customer
  - Recommended old titles
  - Did not respond changes in taste

Collaborative filtering

Customer A:



Customer B:



Customer C:

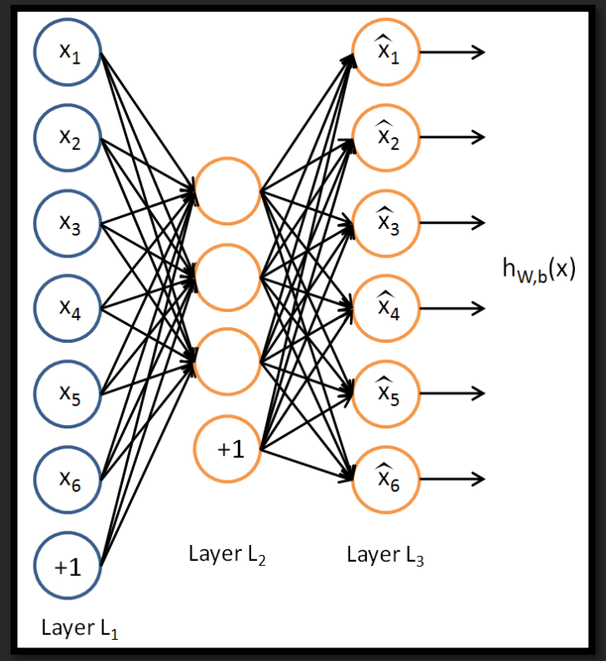


Next steps

- **Solutions?**
  - Feature engineering
  - Linear models on top of embeddings
  - Ensembles
  - Tensor factorization
- **Problem:** Inserting assumptions

Useful insights

- How can we reduce assumptions that might change over time?
  - Deep neural networks
  - Learn representations
- Are we asking the problem in the right way?
  - Insight from Retail Personalization team
  - **Original:** What will the customer eventually watch?
  - **New:** What will the customer watch in the next week?



Neural networks: A simple formulation

Input Layer - X
(2 years till a week ago)



1 Week Ago

Output Layer - Y
(last week)



Neural network: Results

- What worked?
  - Recommendations were appropriate (~3x)
  - Recommendations were comprehensive (~3x)
  - Customer engagement increased significantly (largest increase, at that time)
  - Multi-domain (ICLR 2018 Paper - https://openreview.net/forum?id=S108WDJvf)
- What did not?
  - Some titles felt “out of place”

"When the anecdotes and the data disagree, the anecdotes are usually right."

Jeff Bezos

CEO, Amazon

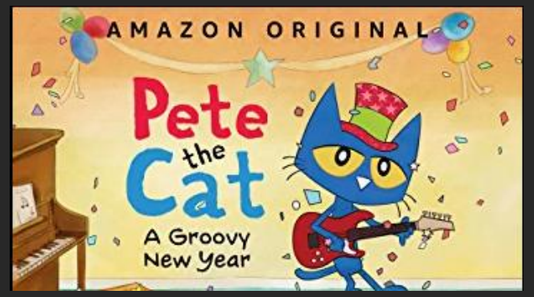
© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Customer trust

- What if the model gets it wrong?
- Customer who watches kids shows:



- Incorrect recommendations:







- Guardrails

So much to explore

- Problem formulations
  - TV watching, completion, diversity
- Different structure
  - Cost functions, activations, layers, optimizers
- Cutting-edge approaches
  - Residual networks, sequence models, attention networks, explore / exploit tradeoffs
- & more

Bottlenecks



- Parameterizing Flows
- AWS Batch / Elastic Inference
- DSSTNE -> MxNet

Simplicity through abstraction

- Hand-crafted features -> Learn features from behavior
- Feature engineering -> Learn the representations from behavior
- Copy & pasting workflows -> Parameterize common steps and flows
- Custom training -> Replace with frameworks as they emerge
- Custom inference -> AWS Batch / AWS Elastic Inference
- Evolving your models -> Amazon Personalize?

In summary

- Start with the customer
- Define success
- Keep it simple, remain nimble
- Embrace failure
- Repeat

Thank you!

Liam Morrison

lmorri@amazon.com

Shaun McCarthy

mcshaun@amazon.com

AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws



Please complete the session survey in the mobile app.

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.