

Recommenders Run the Internet

How to build a recommendation system that drives business growth

Tullie Murrell

Columbia Business School, Oct 2026

Agenda

1. The value of recommendation systems
2. Learning to rank
3. Serving recommendations
4. Evaluating recommenders
5. Generative AI in recommenders

The value of recommendation systems



~80%

of hours watched



~70%

of watch time



~35%

of revenue (est.)



100%

the feed is the product

A model picks the 10 items you see out of a billion. The business depends on which 10.



Story 1: ranking at a social media company

Small modeling changes can turn into millions of dollars.

2. Learning to rank

Predicting order

Regression

Predict a number.

"What will this house sell for?"

Classification

Predict a label.

"Is this email spam?"

Ranking

Predict an order.

"Of 10,000 items, which 10 go first?"

A recommender is the third kind. Its own problem, its own training methods.

Three ways to learn order

	Train on	Asks
Pointwise	One item at a time	Will Ana click this?
Pairwise	Pairs of items	Did Ana prefer A over B?
Listwise	Whole lists	Is this top-10 better than that one?

Most teams start pointwise. Most production systems ended up pairwise.

What goes into the model?

Behavior

Counts and rates.

How many times has this user clicked this item? This category?
What's this item's conversion rate?

Content

User side and item side.

Age, country. Genre, price, description, image.

Context

The request itself.

Device, time of day, where in the app, what they just did.

Example: a pointwise click model

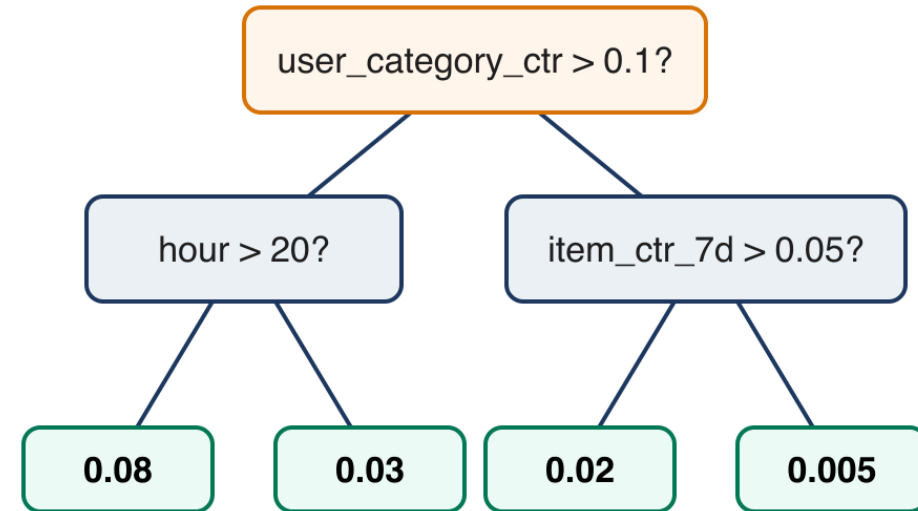
One row: (user, item, context)

behavior user_item_clicks_30d = 0
user_category_ctr = 0.12
item_ctr_7d = 0.04

content item_category = sneakers
item_price = 120
user_country = US

context device = mobile
hour = 22
surface = home_feed

Label clicked = 1 or 0



× hundreds of trees, summed (gradient boosting)

p(click | user, item, context)

score every candidate, sort descending

How do you know it's working?

Classification metrics

AUC, log loss

Is $p(\text{click})$ calibrated? Does it separate clicks from non-clicks?

Ranking metrics

Precision@K, Recall@K, NDCG

Of the top 10 we showed, how many did they click?
Were the clicks near the top?

Hold out last week. Train on everything before. Score the held-out week.

Types of recommendation systems

Collaborative filtering

Learns from behavior only.

"People who clicked this also clicked that." No idea what the item is.

Content-based

Learns from item and user attributes only.

"You liked sneakers under \$150, here are more." No idea what other people did.

Hybrid

Both, plus context.

The tree model two slides ago.
What production systems are.

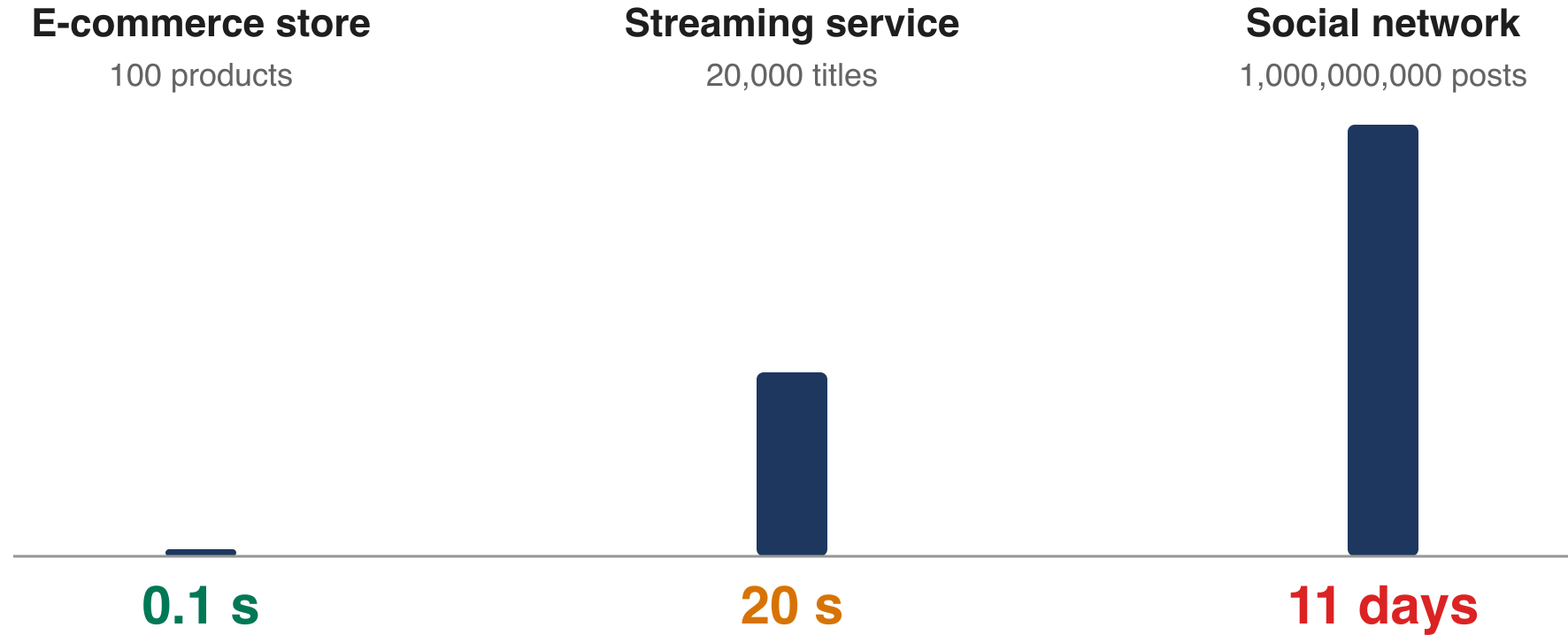


Story 2: stale news at a news publisher

Fixing a model that recommended last year's articles.

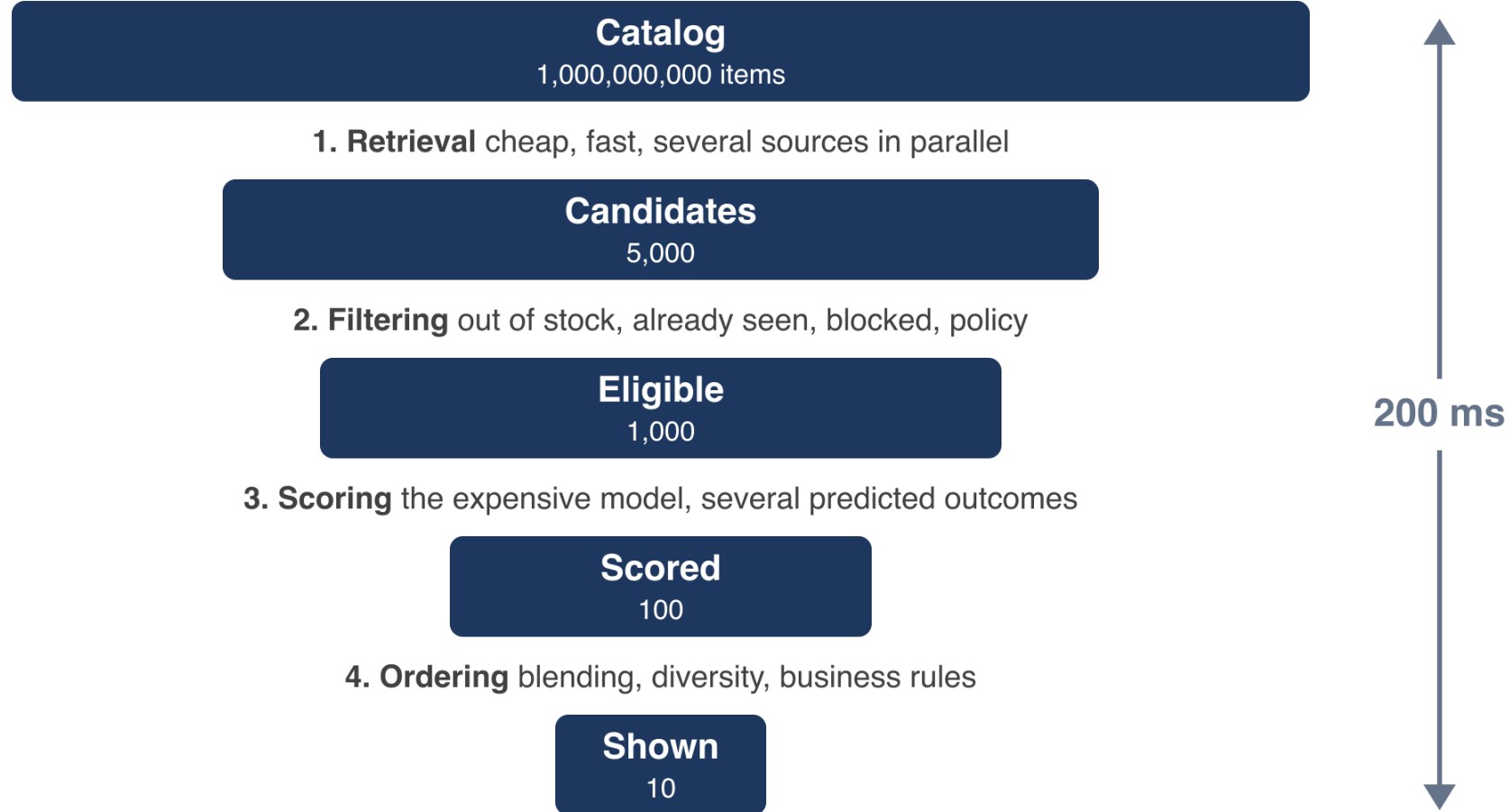
3. Serving recommendations

The latency problem



time to score the whole catalog with a tree model, at 1 ms per (user, item) pair. Log scale.

Multi-stage recommendation systems



The same stages across industries

	Retrieval	Filtering	Scoring	Ordering
Music	Similar songs by nearest-neighbor search	Remove tracks the user has heard	Predict the user will listen to the song	Trade off score, similarity, BPM
Social media	New posts from the user's network	Remove posts from blocked and muted accounts	Predict the user will interact with the post	Reorder so adjacent posts are from different authors
Online store	Items usually co-purchased	Remove out-of-stock items	Predict the user will buy the item	Reorder by price point
Streaming	Items from different rows, shelves, topics	Remove titles unavailable in the user's country	Predict stream time per title	Organize to fit genre distributions

After Oldridge et al., "Building and Deploying a Multi-Stage Recommender System with Merlin", RecSys 2022

Retrieval: several sources, merged

Collaborative filtering

"people like you clicked"

Content filtering

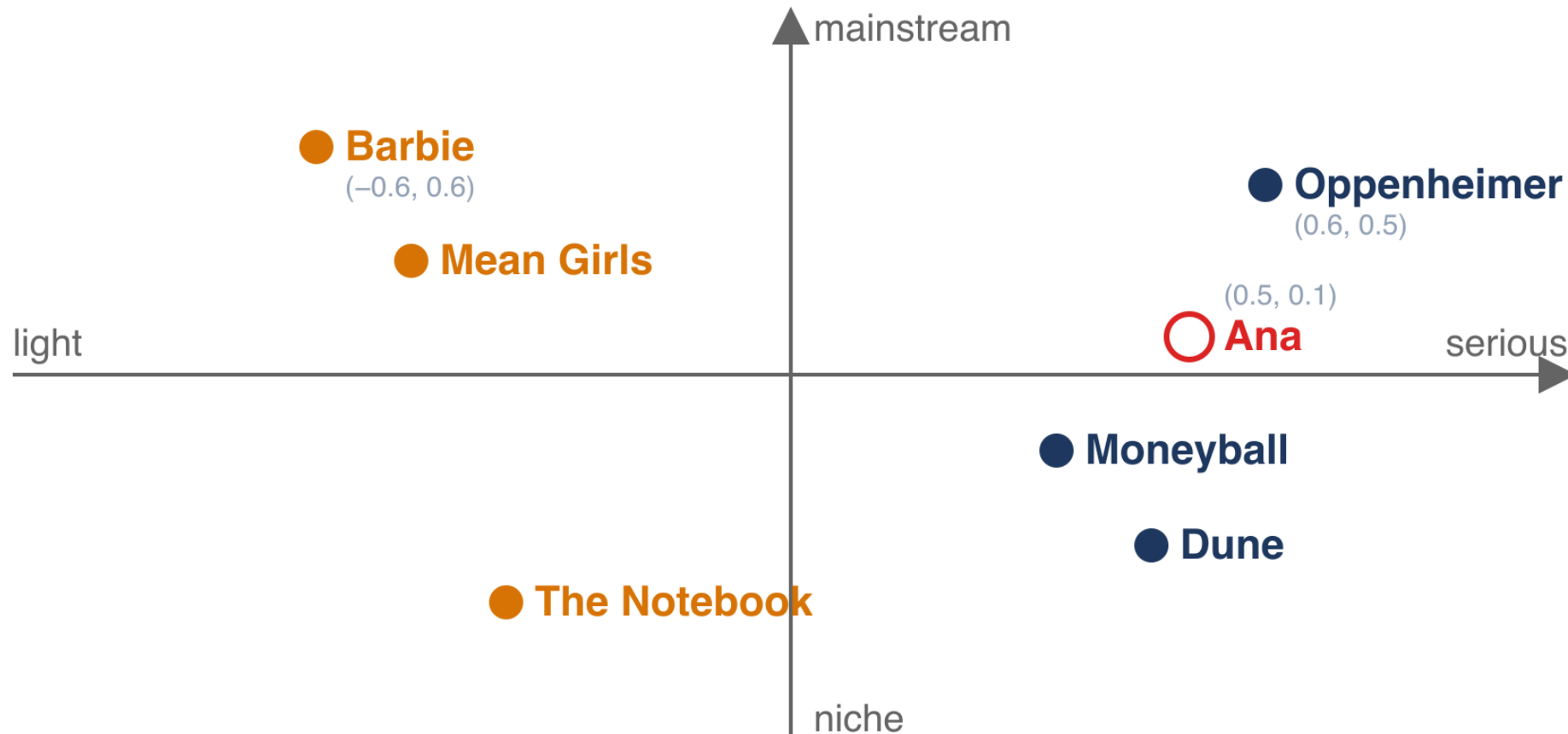
"like what you've clicked"

Heuristics

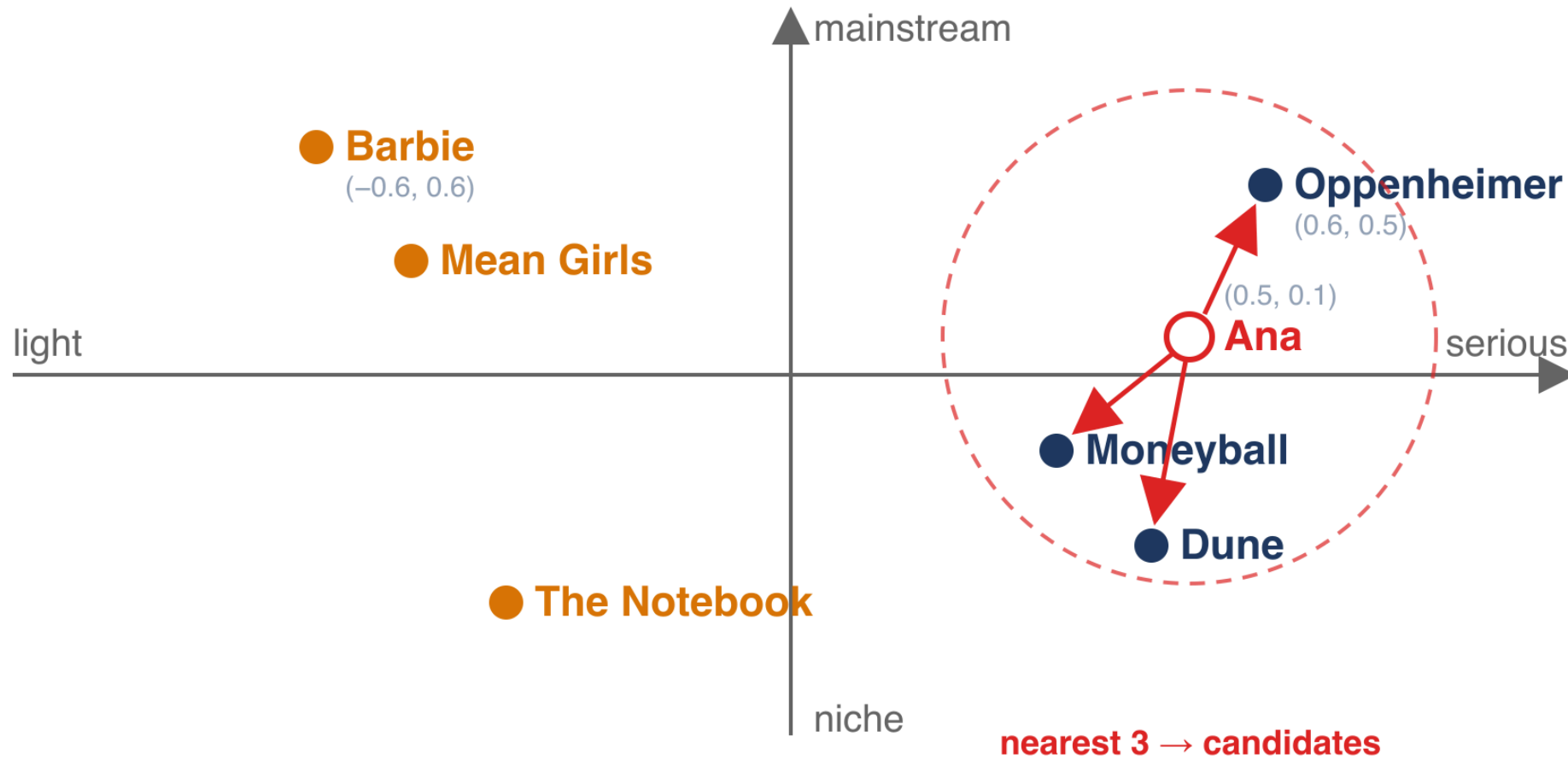
new, trending, from accounts you follow

Job: don't miss anything the scorer would have loved. Order doesn't matter yet.

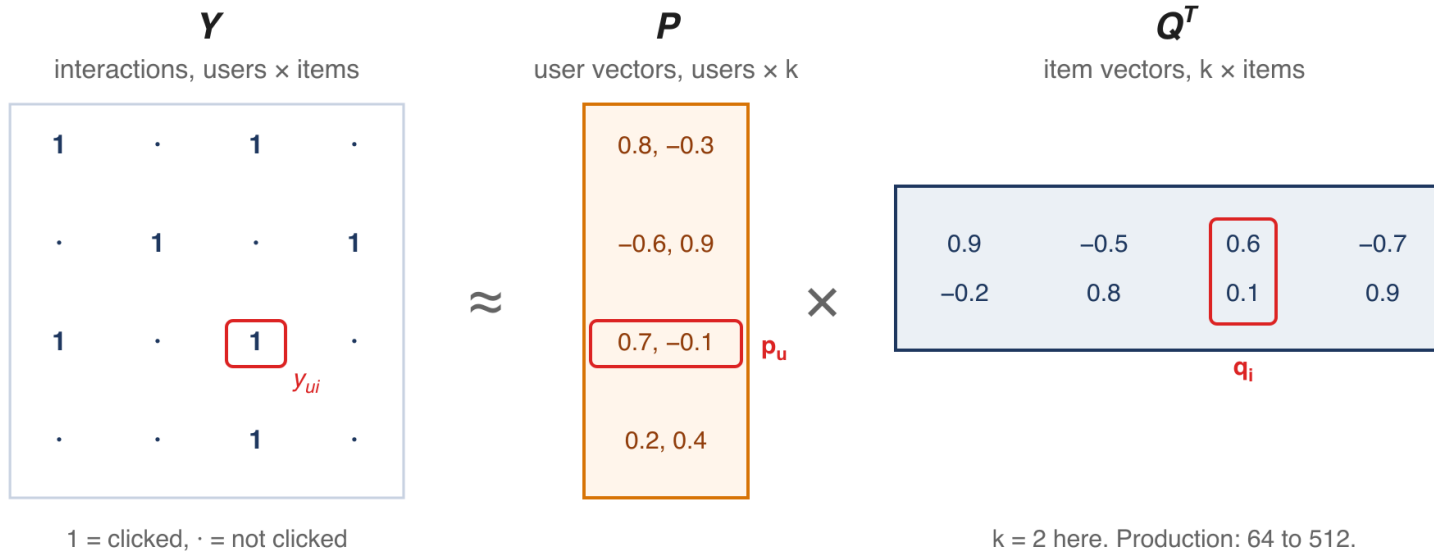
Vector retrieval



Vector retrieval



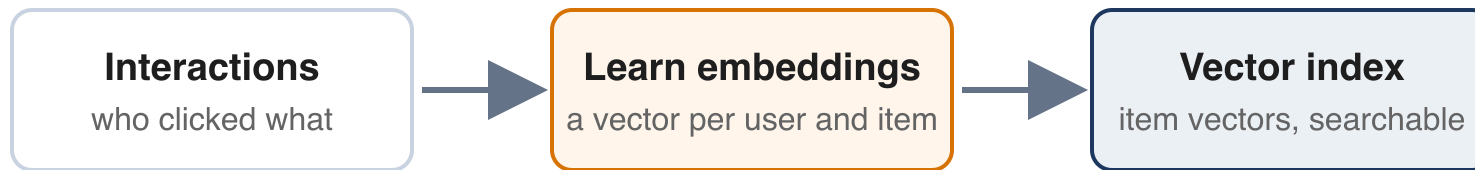
How to build the vectors: matrix factorization



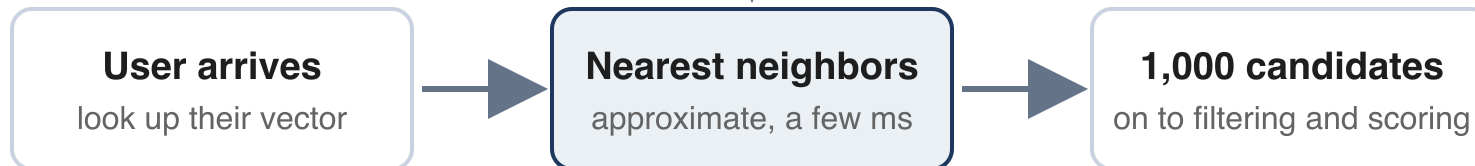
$$Y \approx PQ^T \quad \hat{y}_{ui} = \mathbf{p}_u \cdot \mathbf{q}_i \quad \min_{P,Q} \sum_{(u,i) \text{ observed}} (y_{ui} - \mathbf{p}_u \cdot \mathbf{q}_i)^2 + \lambda(\|\mathbf{p}_u\|^2 + \|\mathbf{q}_i\|^2)$$

Putting it together

OFFLINE, ONCE A DAY



ONLINE, EVERY REQUEST



Story 3: when a stage returns nothing

What do you do when the user has seen everything in the catalog?



Scoring: the ranking model, on the candidates

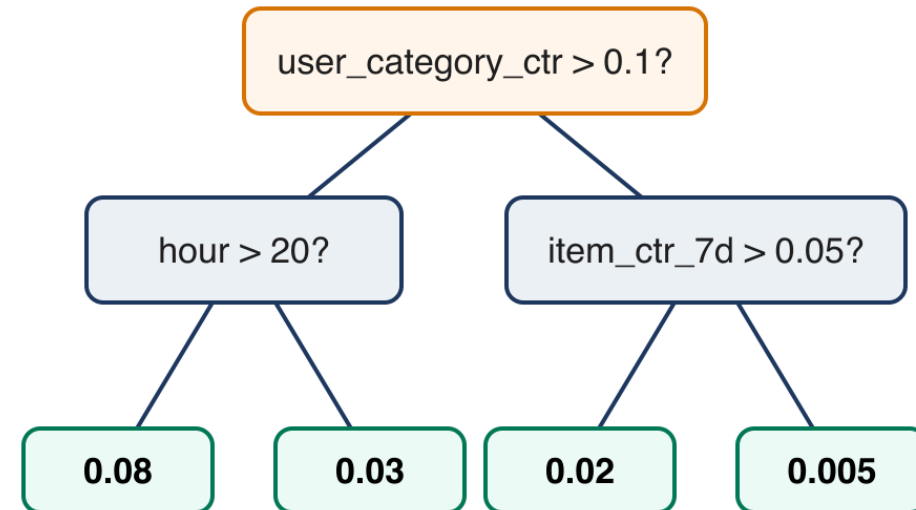
One row: (user, item, context)

behavior user_item_clicks_30d = 0
user_category_ctr = 0.12
item_ctr_7d = 0.04

content item_category = sneakers
item_price = 120
user_country = US

context device = mobile
hour = 22
surface = home_feed

Label clicked = 1 or 0

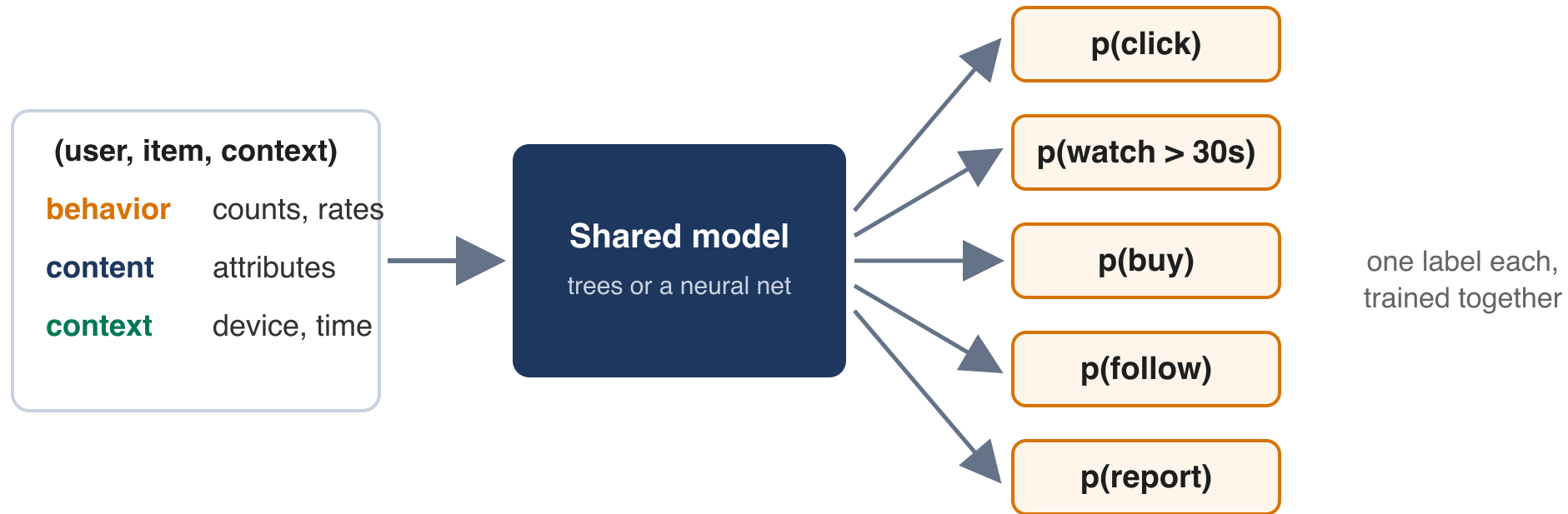


× hundreds of trees, summed (gradient boosting)

p(click | user, item, context)

score every candidate, sort descending

One prediction isn't enough



Combining the predictions

$$\text{score} = w_1 p(\text{click}) + w_2 p(\text{watch}) + w_3 p(\text{buy}) + w_4 p(\text{follow}) - w_5 p(\text{report})$$

The w 's are not learned. People set them.

Story 4: whose metric wins (a social media company)

Every team owned one weight. Raising one lowered another.

Ordering: diversity

By score

Nike Air Max, white	0.31
Nike Air Max, black	0.30
Nike Air Max, grey	0.29
Nike Air Max, red	0.28
Nike Air Max, navy	0.27

five of the same shoe

re-rank



By score, with diversity

Nike Air Max, white	0.31
Adidas Samba	0.22
Running socks	0.19
Nike Air Max, black	0.30
New Balance 574	0.18

lower total score, better page

Rules the scorer can't see:

- no two of the same product in a row
- at most 2 per brand in the top 10
- one item from a new seller
- sponsored slot at position 3

Ordering: exploration

The final 10



9 slots: highest score (exploit)

1 slot: something uncertain (explore)

What goes in the ? slot

- a new item with no data yet
- an item the model is unsure about
- something outside the user's usual

Costs a little engagement now.

It's how the model learns anything new.



Story 5: the filter bubble (a social media company, pre-2017)

What happens when you don't explore.

4. Evaluating recommenders

The A/B test

Offline

Holdout, ranking metrics.

Minutes. Every change. Only sees what the old model showed.

Online

New model to 5% of users. Compare revenue, retention.

Weeks. Noisy. Breaks when users affect each other.

A better model and a better business are different things. Two reasons.

Reason 1: the metric isn't the goal

Clickbait

$p(\text{click}) \uparrow$

$p(\text{come back}) \downarrow$

Autoplay

watch time $\uparrow$

satisfaction $\downarrow$

Popularity bias

offline NDCG $\uparrow$

nothing new gets shown

The offline metric scores what the old system already showed, against what it already got clicked.

Reason 2: on a two-sided platform, users affect each other

Auctions

One more bidder in a thin auction raises the price for everyone in it.

Ride-share, delivery

A driver sent to you is a driver not sent to someone else.

Dating, jobs

A profile shown to you is attention taken from another user.

The click model scores each user alone. The outcome depends on everyone.

Story 6: when online doesn't resolve either (a live-stream marketplace)

A 14-day A/B test often can't see a real +2%.

How to evaluate anyway

Check the offline metric

Did it predict the last ten A/B results? If not, stop using it.

Check the short-term metric

Does it predict retention at 12 months? Re-check every year.

Evaluate on exploration data

So the holdout isn't only what the old model showed.

Run the test long enough

Weeks, not days. Cluster-randomize when users interact.

5. Generative AI in recommenders

Four ways they're being combined

		Status
LLM as feature extractor	Embed descriptions, images, transcripts	Standard. Solves cold start
Generative retrieval	Generate the next item like the next word (Google, 2023)	Early production
Generative recommender	A transformer over the user's action history (Meta, 2024)	In production
LLM as the recommender	Ask ChatGPT what to watch	Good for conversation, not for scale

Where this is going: actions as tokens

LLM

the → cat → sat → on → ?

Generative recommender

watched → skipped → bought → followed → ?

Meta (HSTU) · Netflix · Spotify (reported) · Stripe (transactions, for fraud)

Three things to remember

1. **It's a problem about order, not labels.** Pointwise, pairwise, listwise.
2. **The weights are the strategy.** The model predicts several things. People choose how to combine them.
3. **Offline accuracy is not the goal.** The metric isn't the business. Check that offline wins become online wins.

Thank you

Tullie Murrell