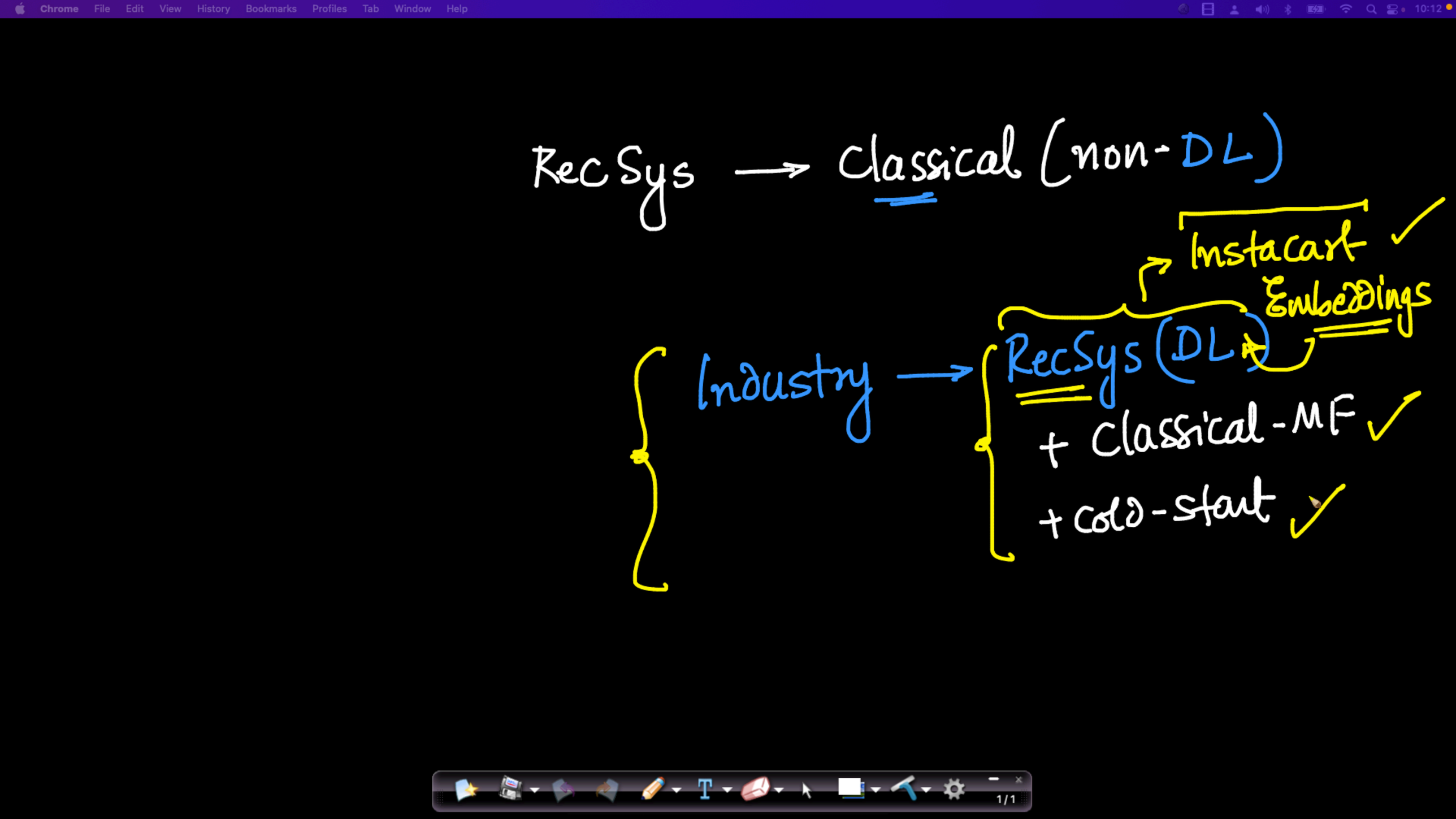

RecSys → Classical (non-DL)
Industry →
RecSys (DL)
Instacart ✓
Embeddings
+ Classical-MF ✓
+ cold-start ✓

Music (Spotify / Gaana / Amazon prime Music / Apple Music / YT Music)

↳ recommend the next few songs to a listner (Task)

Data
|
ML | Science
|
Engg

↳ 100MM Customers [scale]
10MM Songs

→ latency ...

Q

→ lots of new users & Songs...

D1 D2 D3

→ { Mood-specific Songs ↑↑↑...↑ (n)

→ language/region

→ artist/band/director ...

→ 70's/80's/90's songs... (teens/childhood ....)

→

(Q)

How do you measure any RecSys

Metrics

- # skipped Songs ↓ (partial)
- time Spent by the 'user' ↑
- Songs added to playlist / like → explicit (rare)
- ⋮

} implicit (abundant)

Rec-sys-Team 1

Rec-Sys-existing

A/B testing

Split by users?

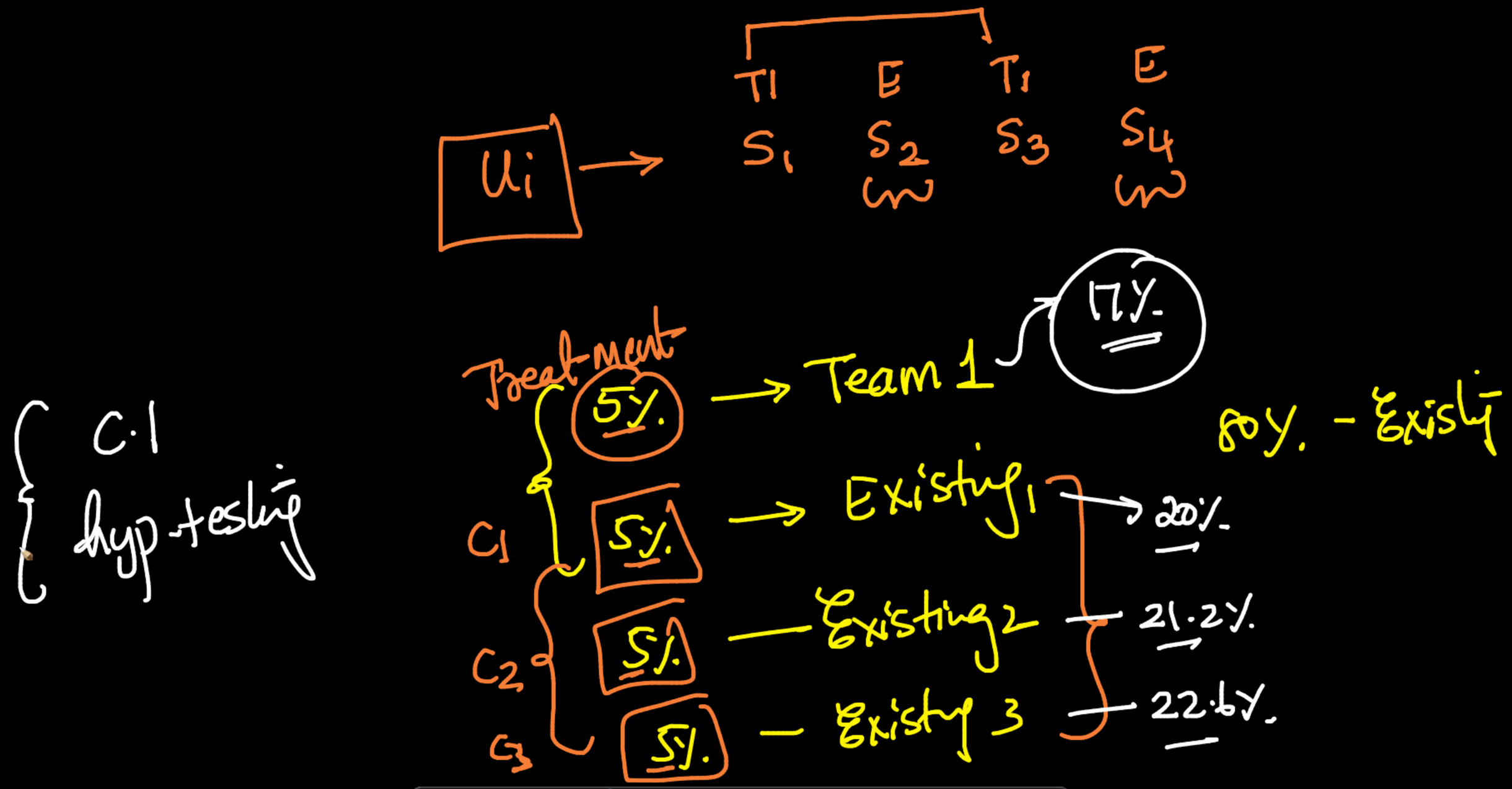
(req. for a song)

→ Ratio: Team1: 5%
Existing: 95%

User

user's req for a song ✓

✓


Ui
T1 E T1 E
S1 S2 S3 S4
17%
Treatment
5%
Team 1
80% - Existing
C1
5%
Existing 1
20%
C2
5%
Existing 2
21.2%
C3
5%
Existing 3
22.6%
C.I
hyp-testing

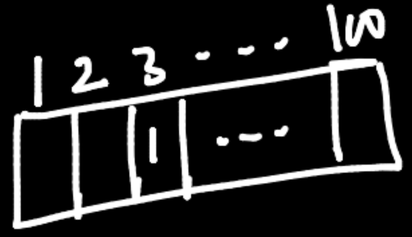
Chrome File Edit View History Bookmarks Profiles Tab Window Help 10:39

[Causal Inference → research . . .

A/B testing

7 / 8

Q



20:

- ocassion → festival / .....
  - age / location
- user → features sign up form, ...
- Song → Title, artist, ...

- t_k, u_i, s_j → historical data
  - liked / playlist (0/1)
  - skipped (0 - 1)

Q

{ User - features
  Song - features

$f(u, s) \rightarrow p(u_i \text{ likes } s_j)$

ignoring t_k, u_i, s_j info

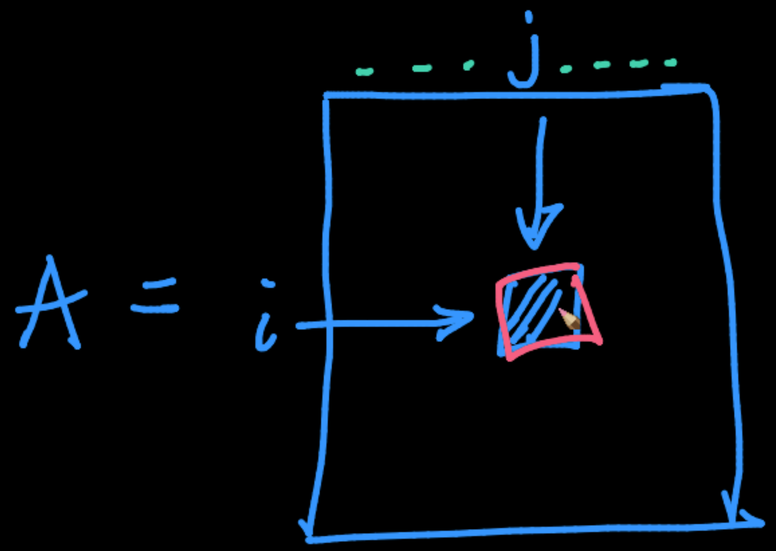
Recap:

- → Problem-def & challenges/nuances
- → Metrics & A/B testing
- → Data labels

Team 1 →

Models

① Collaborative filtering : MF



(u_i, s_j) → [ ] → #times the user$_i$ played song$_j$

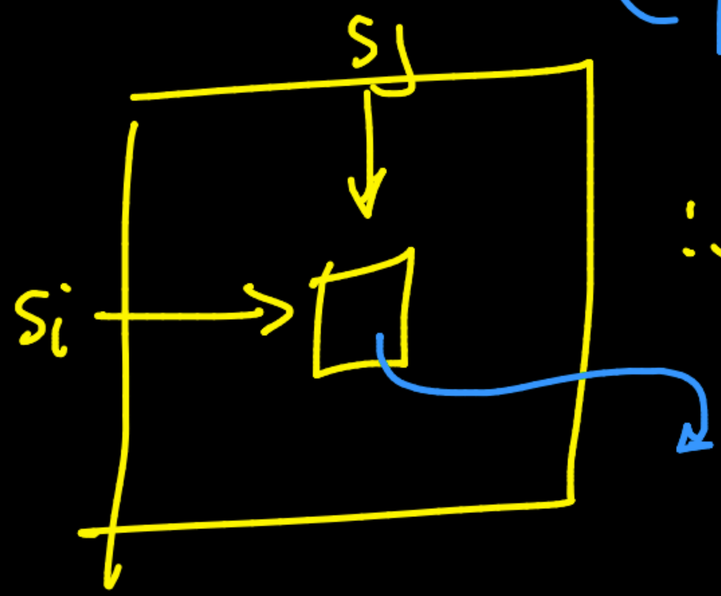
-1 / 1 → liked/played

skipped

0 → no-history

(a) time → s_i after s_j (billions of these pairs)

(b)



1

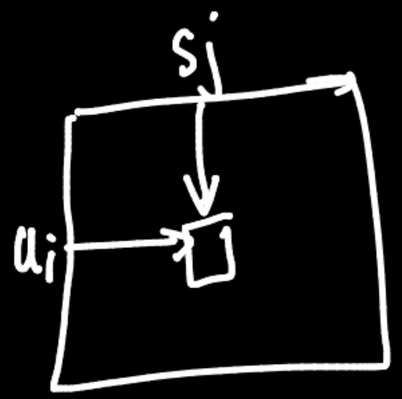
t_k, u_i, s_j

→ liked ; played ; skipped ; bought ; .... (many actions)

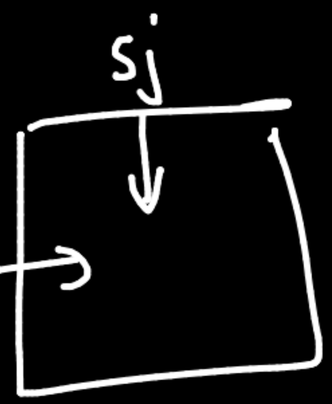
Multiple matrices :

2

L:



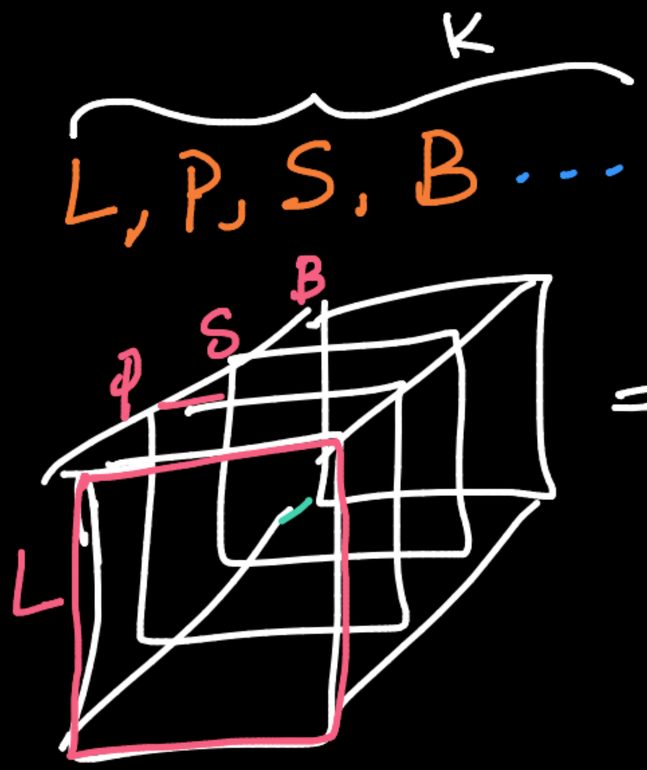
P:



S

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Matrices: L, P, S, B ....



$= A_{n \times m \times k} = \theta_{1 \times d} \cdot X_{d \times 1}$

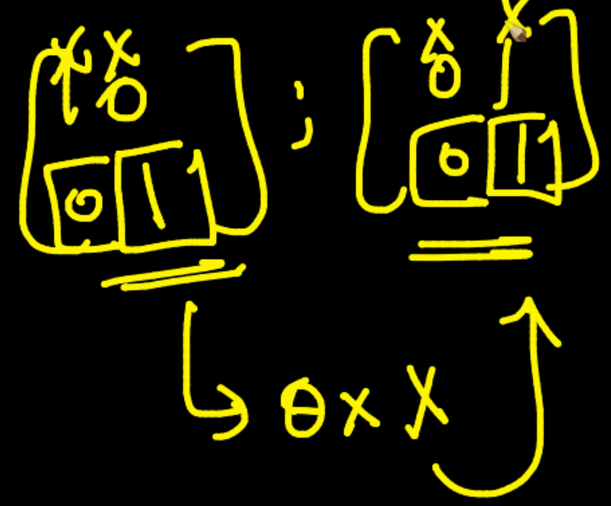
binary

$L = \theta_{n \times d} \cdot X_{d \times m}$

X

$P = \theta_{n \times d} \cdot X_{d \times m}$

real value 0 - 1



Typical MF: $A_{n \times m} = B_{n \times d} : C_{d \times m}$

time!

$L_{n \times m \times 30 \cancel{365}}$

Real-world

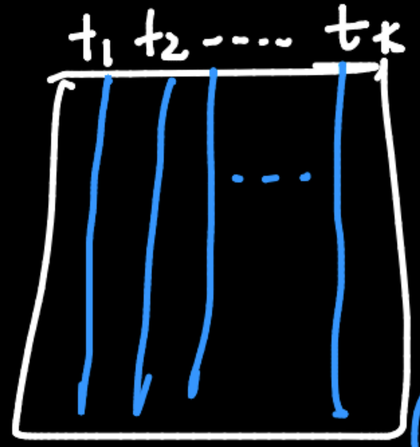
Hybrid → Content-based + coll. filtering

cold-start

2D

① time

L : n x m x k



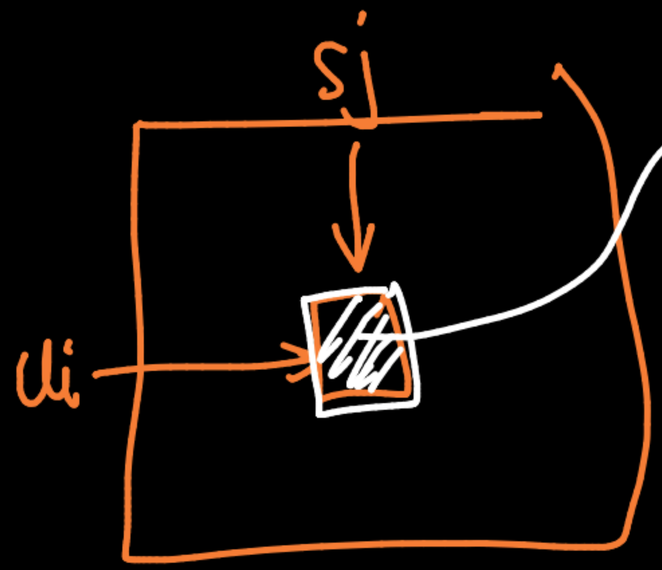
(n x m) x K

② L, R, S, ...

Score: 1 2 0

0 0 1

(random)



$n \times m = N$

$L_{N \times K} = B_{N \times d} \cdot C_{d \times K}$

each row/col represent?

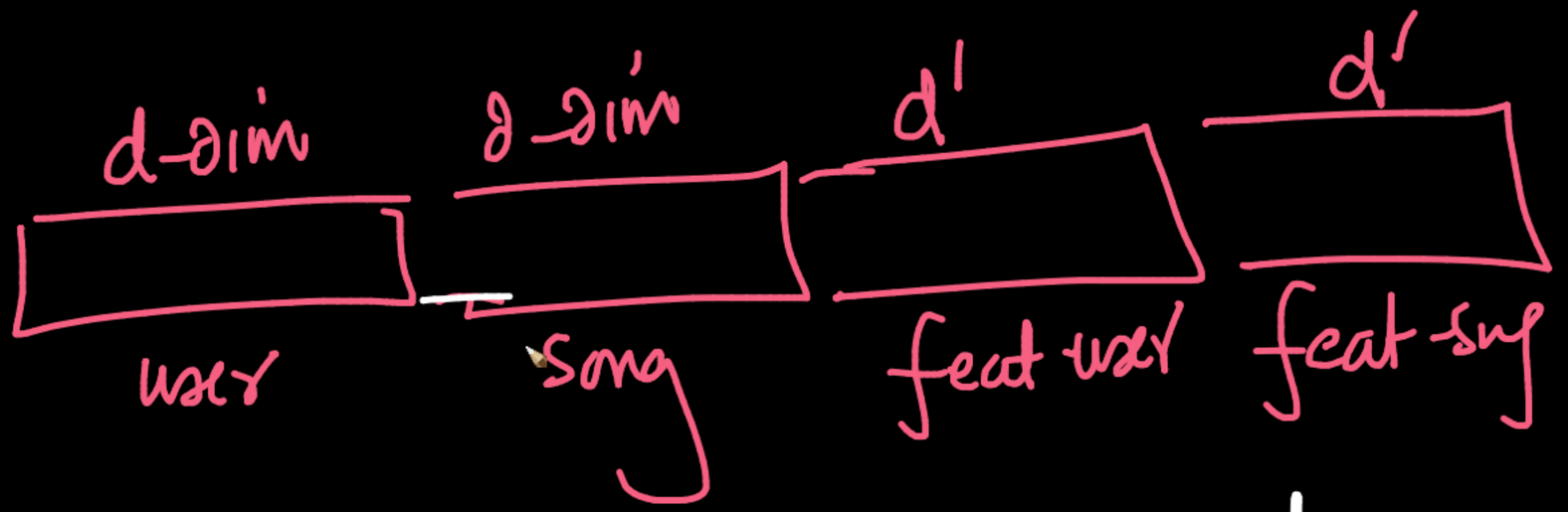
Variations across time per (u_i, s_j)

$A_{ij} = B_i \cdot C_j$

$A_{n \times m} = B_{n \times d} \; C_{d \times m}$

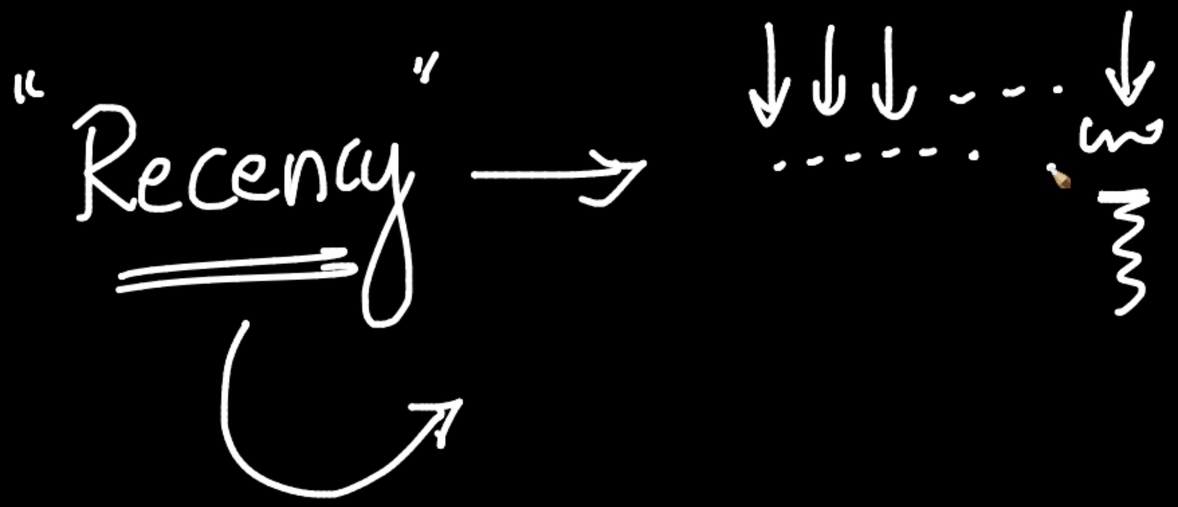
each row: user ✓

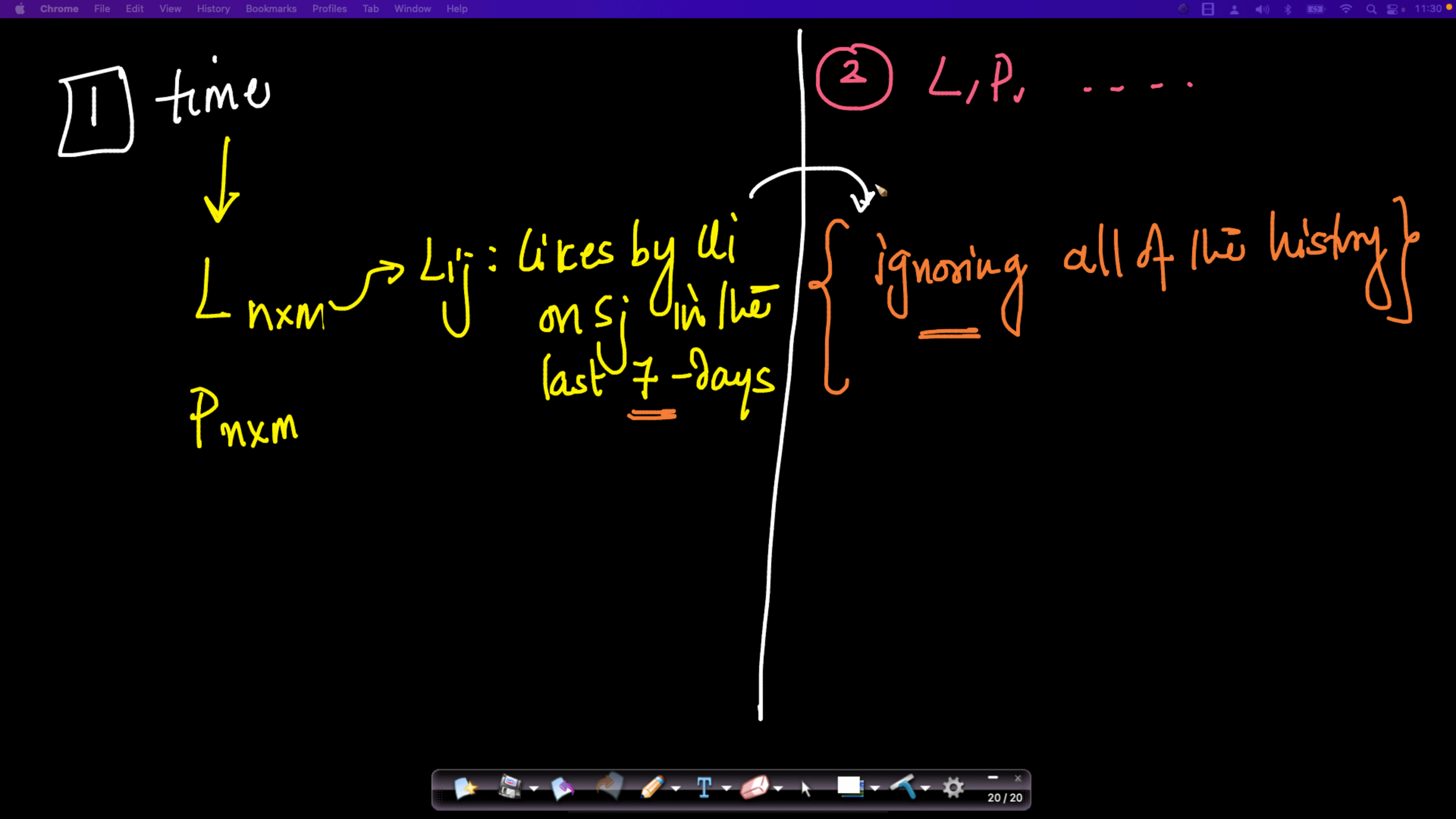
each col: song ✓

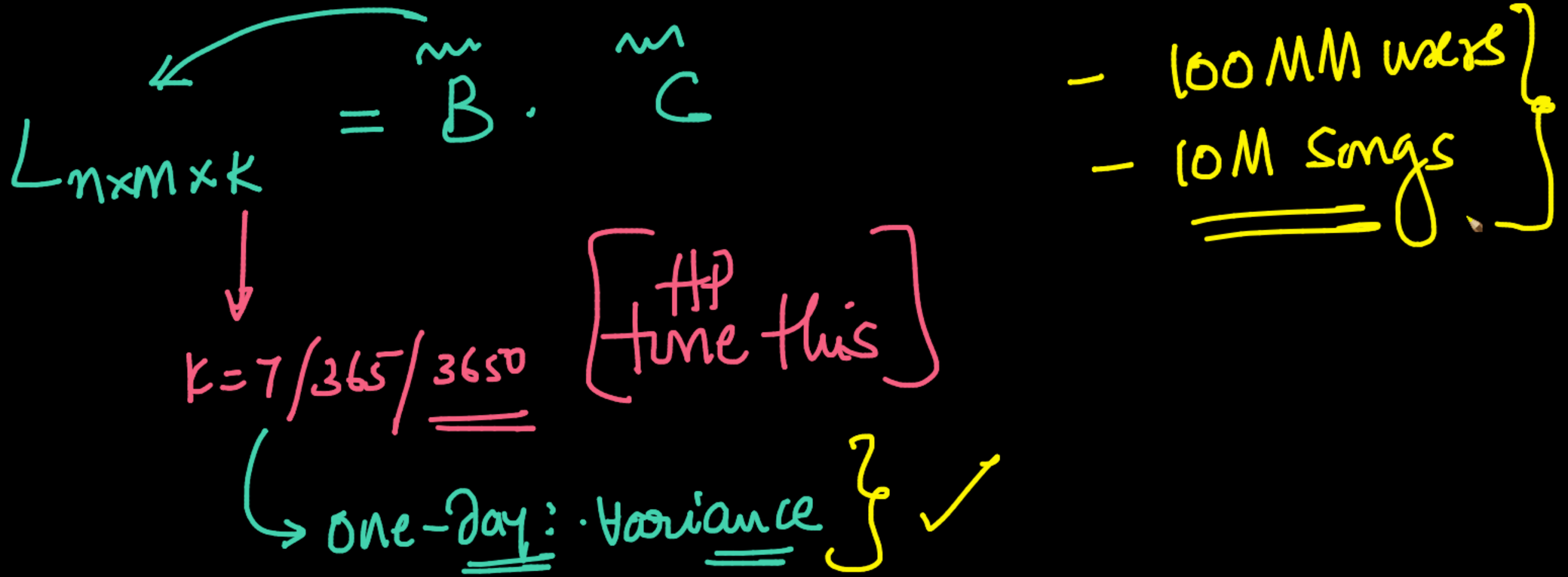


Non-linear model

L/P/S - -


"Recency"


1 time
L nxm
Lij : likes by ui on sj in the last 7 -days
P nxm
2 L, P, ...
{ ignoring all of the history }


$L_{n \times m \times k}$
$= B \cdot C$
- 100 MM users
- 10M songs
K = 7 / 365 / 3650
[Hp time this]
one-day : Variance ✓

next-session → continuation

- MF : time & multiple-signals (L, P, S ...) → i-i & u-u similarity ... {Col. Filtering}
- scale of MF (training) → Interpretability
- recency & historical info → retrain
- cold start → item-item ... →
- productionize (latency ; cost)

Recap:

1. Defined the problem; scale; key aspects
2. Metrics; A/B testing (briefly)
3. Data: $t_k, u_i, s_j \rightarrow$ L/Pl/S/... (actions)
4. MF & Col-fi Hexing
   - → time
   - → actions

- → Tensor fact X
- → Tall&Skinny Matr X
- → scoring X

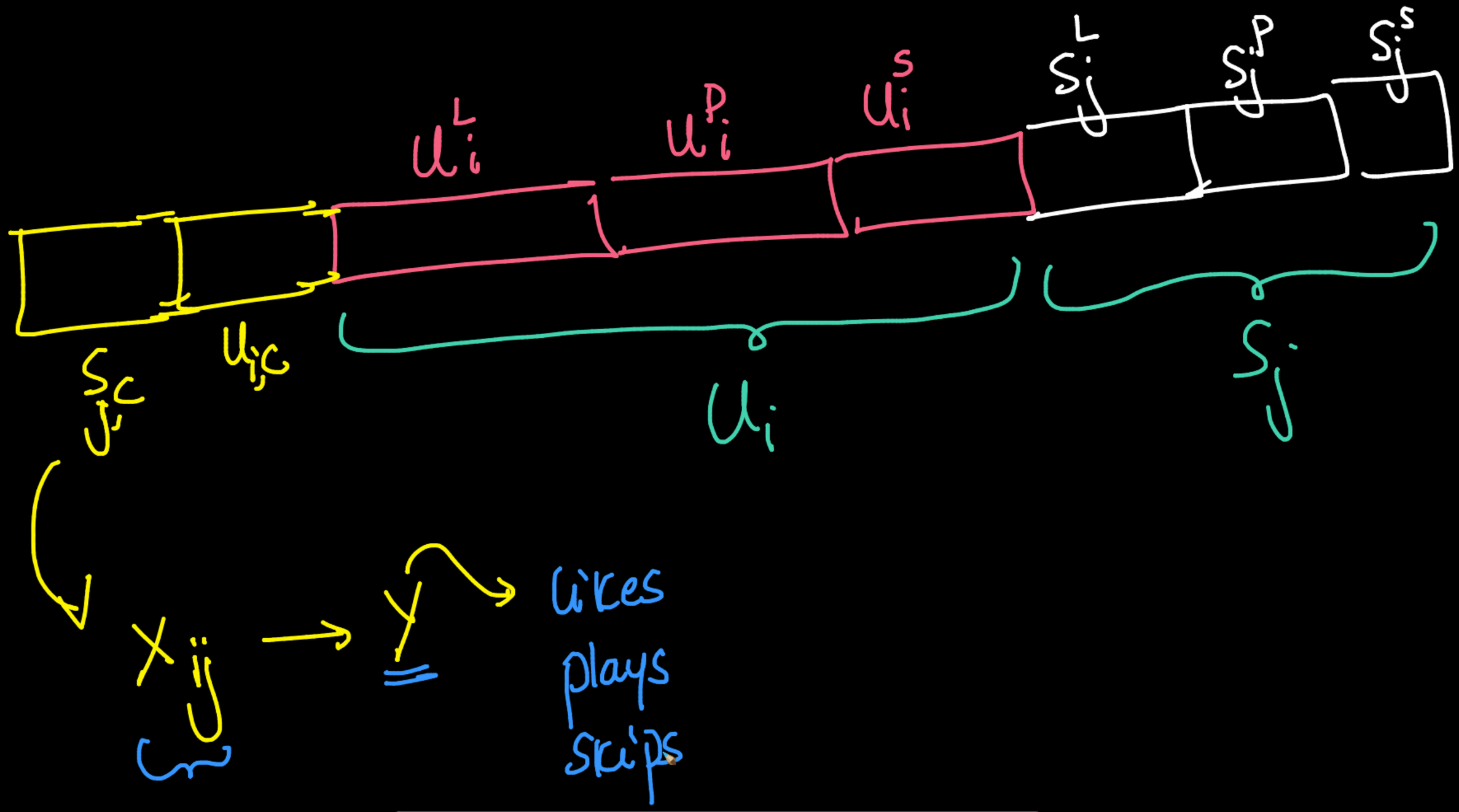
Good-idea

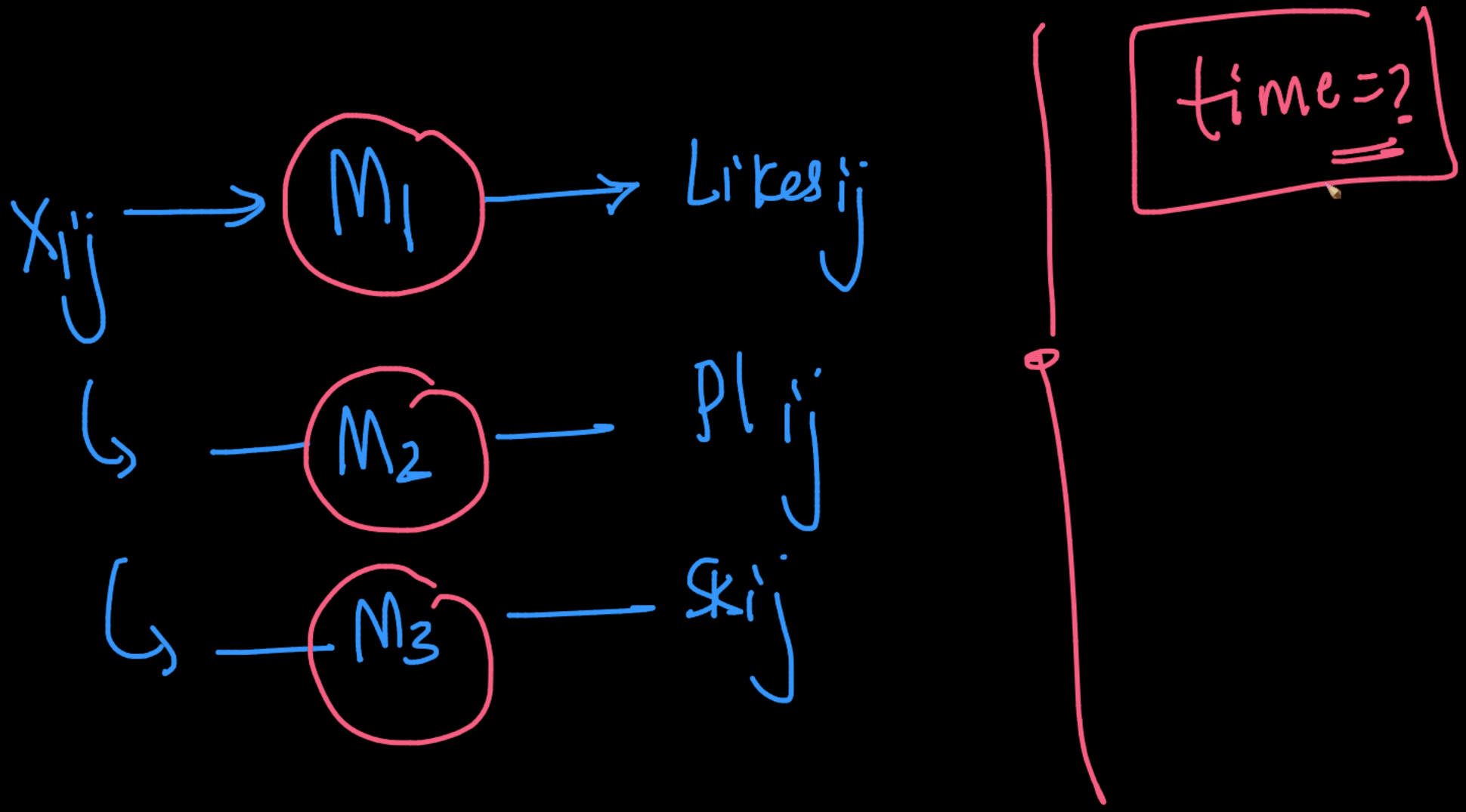
$$L_{n\times m} = A^{(1)}_{n\times d}\, B^{(1)}_{d\times m} \quad \therefore u_i^{(1)}\, s_j^{(1)}$$

$$P_{n\times m} = A^{(2)}_{n\times d}\, B^{(2)}_{d\times M} \quad u_i^{(2)}\, s_j^{(2)}$$

$$S_{n\times m} = A^{(3)}_{n\times d}\, B^{(3)}_{d\times m} \quad u_i^{(3)}, s_j^{(3)}$$

higher weightage to L


u_i^L
u_i^P
u_i^S
s_j^L
s_j^P
s_j^S
$s_{j,c}$
$u_{i,c}$
u_i
s_j
x_{ij}
likes
plays
skips


x_{ij}
M_1
Likes ij
M_2
Pl ij
M_3
Sk ij
time =?

Time

$\{L_{n \times m \times 28}$ → historical taste (weighted by time)

↳ 7 days x 4 (6-hrs)

Factorize this?

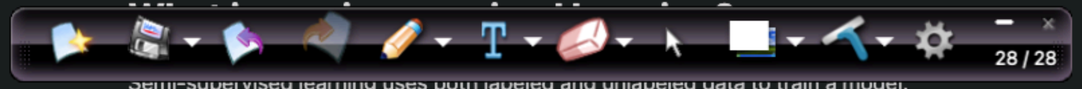
Learning with not Enough Data Part 1: Semi-Supervised Learning

December 5, 2021 · 26 min · Lilian Weng

When facing a limited amount of labeled data for supervised learning tasks, four approaches are commonly discussed.

1. **Pre-training + fine-tuning**: Pre-train a powerful task-agnostic model on a large unsupervised data corpus, e.g. pre-training LMs on free text, or pre-training vision models on unlabelled images via self-supervised learning, and then fine-tune it on the downstream task with a small set of labeled samples.
2. **Semi-supervised learning**: Learn from the labelled and unlabeled samples together. A lot of research has happened on vision tasks within this approach.
3. **Active learning**: Labeling is expensive, but we still want to collect more given a cost budget. Active learning learns to select most valuable unlabeled samples to be collected next and helps us act smartly with a limited budget.
4. **Pre-training + dataset auto-generation**: Given a capable pre-trained model, we can utilize it to auto-generate a lot more labeled samples. This has been especially popular within the language domain driven by the success of few-shot learning.

I plan to write a series of posts on the topic of "Learning with not enough data". Part 1 is on *Semi-Supervised Learning*.



Semi-supervised learning uses both labeled and unlabeled data to train a model.

Interestingly most existing literature on semi-supervised learning focuses on vision tasks. And instead pre-training + fine-tuning is a more common paradigm for language tasks.