Recommendation Systems

Using Deep Learning Techniques

9th & 10th Sept 2019
Workshop Notes

AI Conference 2019
San Jose, CA

Amit & Bargava

Agenda - Day 1

- Workshop Introduction
- #1: Recommendation Framework
- #2: Deep Learning Basics
- #3: Collaborative Filtering
- #4: Content Based
- #5: Learning to Rank
- Recap + Q&A

Agenda - Day 2

- Introduction + Q&A
- #5: Learning to Rank (contd.)
- #6: Hybrid Recommender
- #7: Time & Context
- #8: Deployment & Monitoring
- #9: Evaluation & Challenges
- Recap & Way Forward

Session Plan

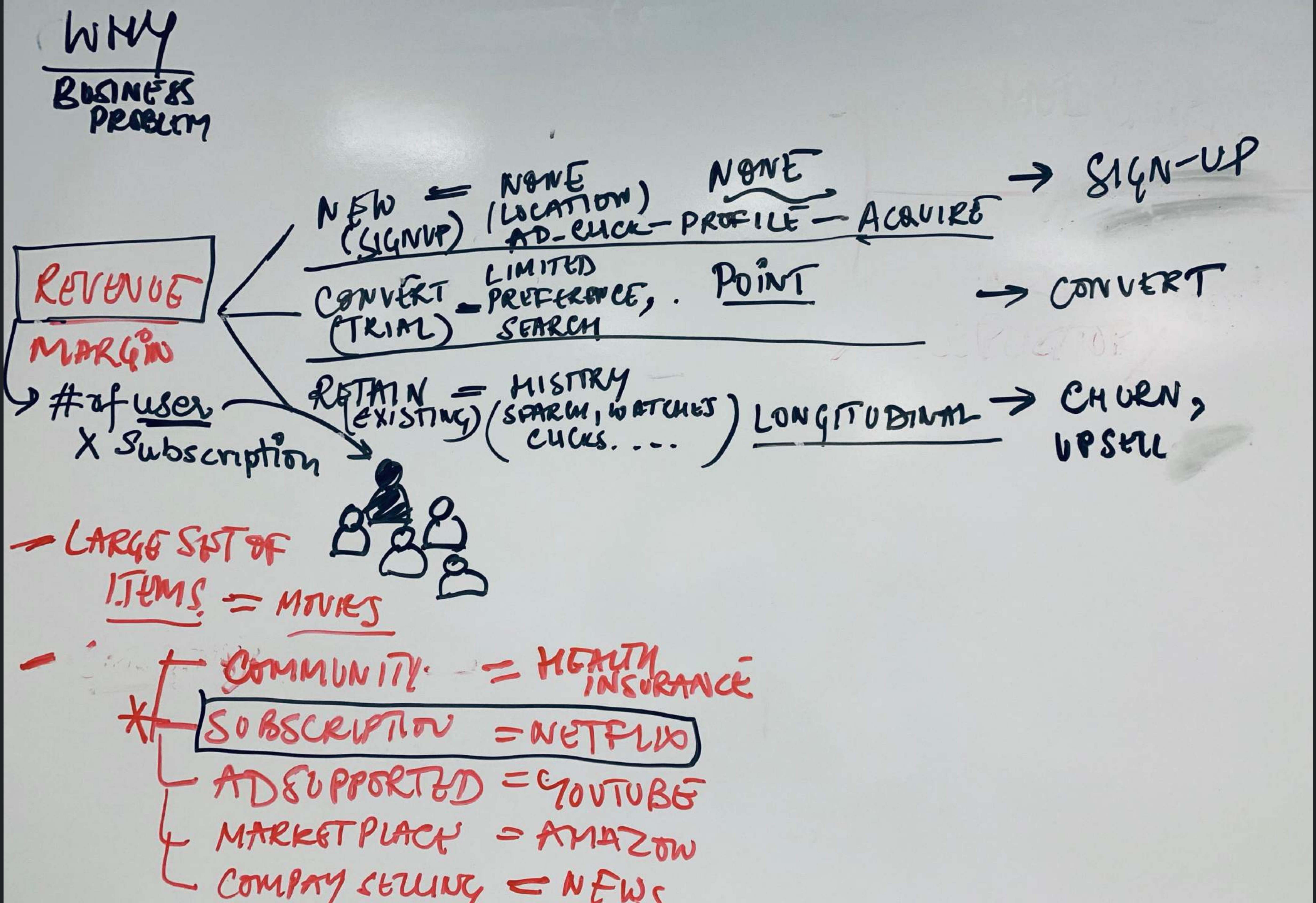
— 0930 - 1030: Session #1
— 1030 - 1100: Morning Break
— 1100 - 1230: Session #2
— 1230 - 1330: Lunch
— 1330 - 1500: Session #3
— 1500 - 1530: Afternoon Break
— 1530 - 1700: Session #4

RecSys Process

- **Why**: Define (Business Problem)
- **What**: Frame (User-Item-Interaction)
- **How**:
  - Data: Acquire, Refine, Transform
  - Visual Explore
  - Model: Build, Evaluate, Tune and Select
- **So What**: Show & Serve
- **So Why**: Measure, Test, Improve

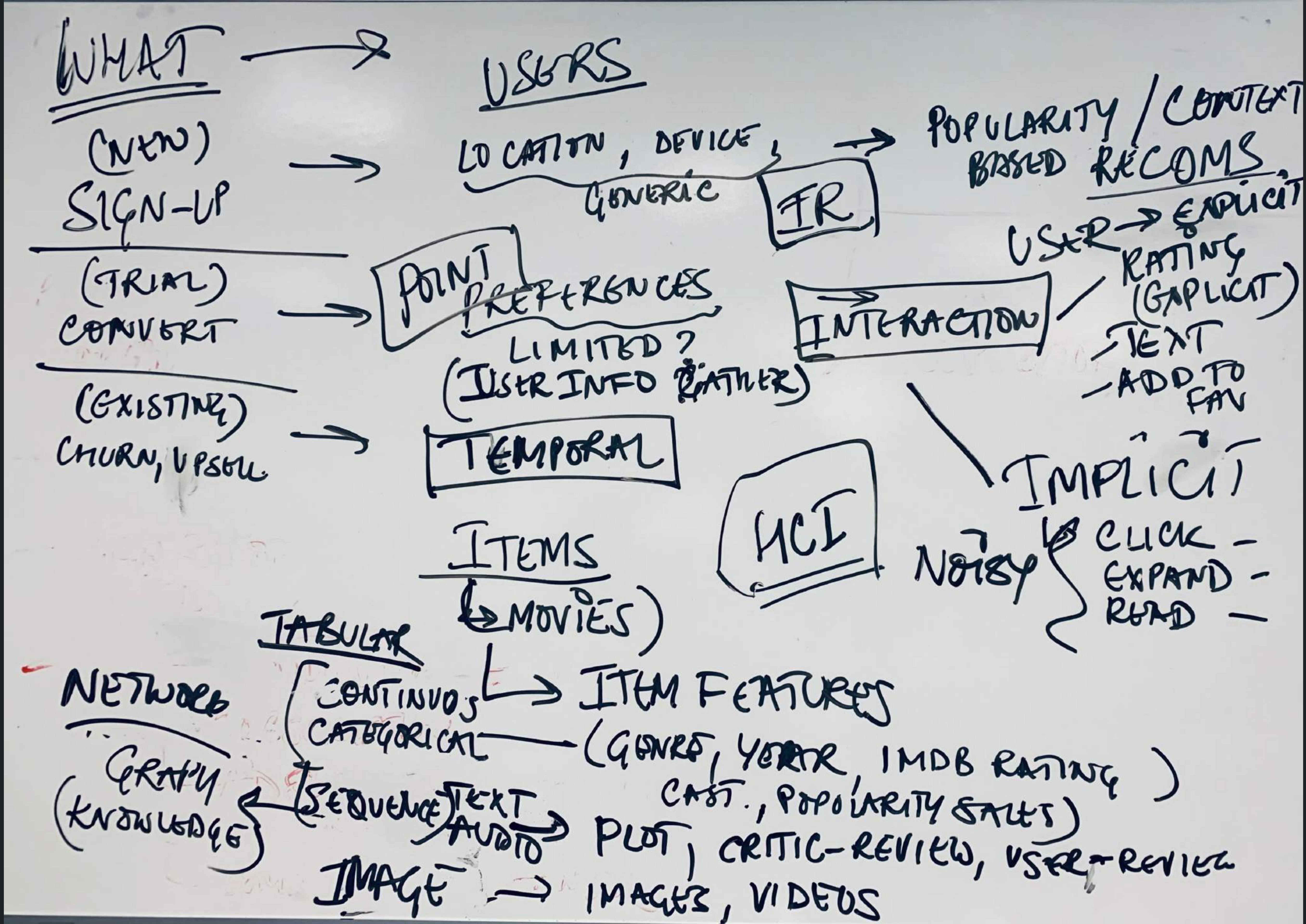
Why Recommendations

- Type of Companies (Subscription, Ad-Supported, Marketplace, Company Site, Community)
- RecSys objectives: Revenue or Margin based
  - **New**: Discovery, Sign-ups
  - **Convert**: Trial to Conversion
  - **Retain**: Reduce Churn, Up-sell, Discovery


WHY
BUSINESS PROBLEM
REVENUE
MARGIN
# of user X Subscription
NEW (SIGNUP) = NONE (LOCATION) AD_CLICK - PROFILE - ACQUIRE
NONE
→ SIGN-UP
CONVERT (TRIAL) = LIMITED PREFERENCE, SEARCH
POINT
→ CONVERT
RETAIN (EXISTING) = HISTORY (SEARCH, WATCHES CLICKS. . .)
LONGITUDINAL → CHURN, UPSELL
- LARGE SET OF ITEMS = MOVIES
- COMMUNITY = HEALTH INSURANCE
* SUBSCRIPTION = NETFLIX
ADSUPPORTED = YOUTUBE
MARKETPLACE = AMAZON
COMPANY SELLING = NEWS

What Recommendations

- Inputs
  - Users vs. Items
  - Interaction
  - User & Item Features
- Outputs
  - User & Item Representation
  - Prediction Function
  - Ranking Function


WHAT
USERS
(NEW) SIGN-UP
LOCATION, DEVICE, GENERIC
FR
POPULARITY / CONTEXT BASED RECOMS
USER → EXPLICIT
RATING (EXPLICIT)
TEXT
ADD TO FAV
(TRIAL) CONVERT
POINT PREFERENCES
LIMITED?
(USER INFO GATHER)
INTERACTION
(EXISTING) CHURN, UPSELL
TEMPORAL
HCI
IMPLICIT
NOISY
CLICK
EXPAND
READ
ITEMS
(MOVIES)
TABULAR
CONTINUOUS
CATEGORICAL
NETWORK
GRAPH (KNOWLEDGE)
SEQUENCE
TEXT
AUDIO
IMAGE
ITEM FEATURES
(GENRE, YEAR, IMDB RATING, CAST, POPULARITY SALES)
PLOT, CRITIC-REVIEW, USER-REVIEW
IMAGES, VIDEOS

Users vs. Items

Users
A user is the active party, receiving and acting on the recommendations.
(Can be the context, not an actual person!)

Items
An item is the passive party that is being recommended to the users.

Interaction Actions: Positive vs. Negative

Positive
Favourites, likes, stars, watches, shares, follows, bids, purchases, reads, views, upvotes…

Negative
Downvote, skip, 1-star reviews, rejections, unfollows, returns, downvotes…

Interaction Types: Explicit vs Implicit

Explicit
Actions that a user expects or intends to impact their personalized experience.

Implicit
All other actions observed as interactions between users and items.

Interaction Scope: Point vs Temporal

Point interaction consolidate all the interaction between a user and item into a single number e.g. Rating info

Longitudinal interactions retain the history of interaction over timestamps e.g. Session-based

User & Item Features: Indicator

Indicator Features
- Feature unique to every user/item to allow for direct personalization.
- Allow to learn about every user individually without being diluted through metadata.
- Often one-hot encoded user IDs or just an identity matrix.

User & Item Features: Metadata

Metadata Features

- Types of features
  - **Continuous or Categorical**: Age, location, language, device, watch time, ...
  - **Sequence-based**: tags, labels, word counts, audios
  - **Image or Video Based**: posters, videos, trailers, ...
  - **Graph-based**: knowledge base about items
- Every element about user/item before training can be a feature (if properly structured)
- Often called "side input" or "shared features."

How Recommendations

- Traditional Approaches
  - Top-N (Popularity based Similarity)
  - Similar Items (Content Based Similarity)
  - Frequent Item Set (Transaction Based Similarity)
  - Personalisation (User Based Similarity)
  - Collaborative Filters (Latent Features)
- Deep Learning Based Approaches

Approaches to build a Recommendation System

Fundamental Building Blocks

Items : Object to be recommended
Users : Target of the recommendation
Signal: Explicit or Implicit feedback

Items = Movies

items	WallE	iRobot	Cars	Shrek	RoboCop

A

Views	8k	5k	6k	3k	1k
items	WallE	iRobot	Cars	Shrek	RoboCop

Top-N Items
(Popularity based Ranking)

B

Rating	8.4	7.1	7.1	7.9	6.2
Genre	Anim	SciFi	Anim	Anim	SciFi
Views	8k	5k	6k	3k	1k
items	WallE	iRobot	Cars	Shrek	RoboCop

Similar Items

(Content based Approach)

C

Rating	8.4	7.1	7.1	7.9	6.2
Genre	Anim	SciFi	Anim	Anim	SciFi
Views	8k	5k	6k	3k	1k

Trans	WallE	iRobot	Cars	Shrek	RoboCop
#001	1		1		
#002		1			1
#003	1			1	
#004			1		

Frequent Item Set
(Transaction based Similarity)

D

Rating	8.4	7.1	7.1	7.9	6.2
Genre	Anim	SciFi	Anim	Anim	SciFi
Views	8k	5k	6k	3k	1k

age	loc	Users	WallE	iRobot	Cars	Shrek	RoboCop
16	IN	Ela	1		1		
23	IN	Bhim		1			1
35	SG	Chan	1			1	
29	SG	Dan			1		

Personalisation
(User Demographic based Similarity)

User-Item Latent Factor
(Collaborative Filtering)

	WallE	iRobot	Cars	Shrek	RoboCop
Rating	8.4	7.1	7.1	7.9	6.2

age	loc		WallE	iRobot	Cars	Shrek	RoboCop	lfi1	lfi2
16	IN	Ela	1		1			0.51	0.76
23	IN	Bhim		1			1	0.54	0.94
35	SG	Chan	1			1		0.78	0.64
29	SG	Dan			1			0.44	0.93

	WallE	iRobot	Cars	Shrek	RoboCop
lfu1	0.45	0.76	0.84	0.83	0.36
lfu2	1.01	0.63	0.69	0.51	0.83

F

	WallE	iRobot	Cars	Shrek	RoboCop
Rating	8.4	7.1	7.1	7.9	6.2
Genre	Anim	SciFi	Anim	Anim	SciFi
Views	8k	5k	6k	3k	1k

age	loc
16	IN
23	IN
35	SG
29	SG

	WallE	iRobot	Cars	Shrek	RoboCop
Ela	1		1		
Bhim		1			1
Chan	1			1	
Dan			1		

Tensor-based

(Deep Learning Models)

“In theory, there is no difference between theory and practice. In practice, there is.”

Yogi Berra

"In theory, there is no difference between theory and practice. In practice, there is."

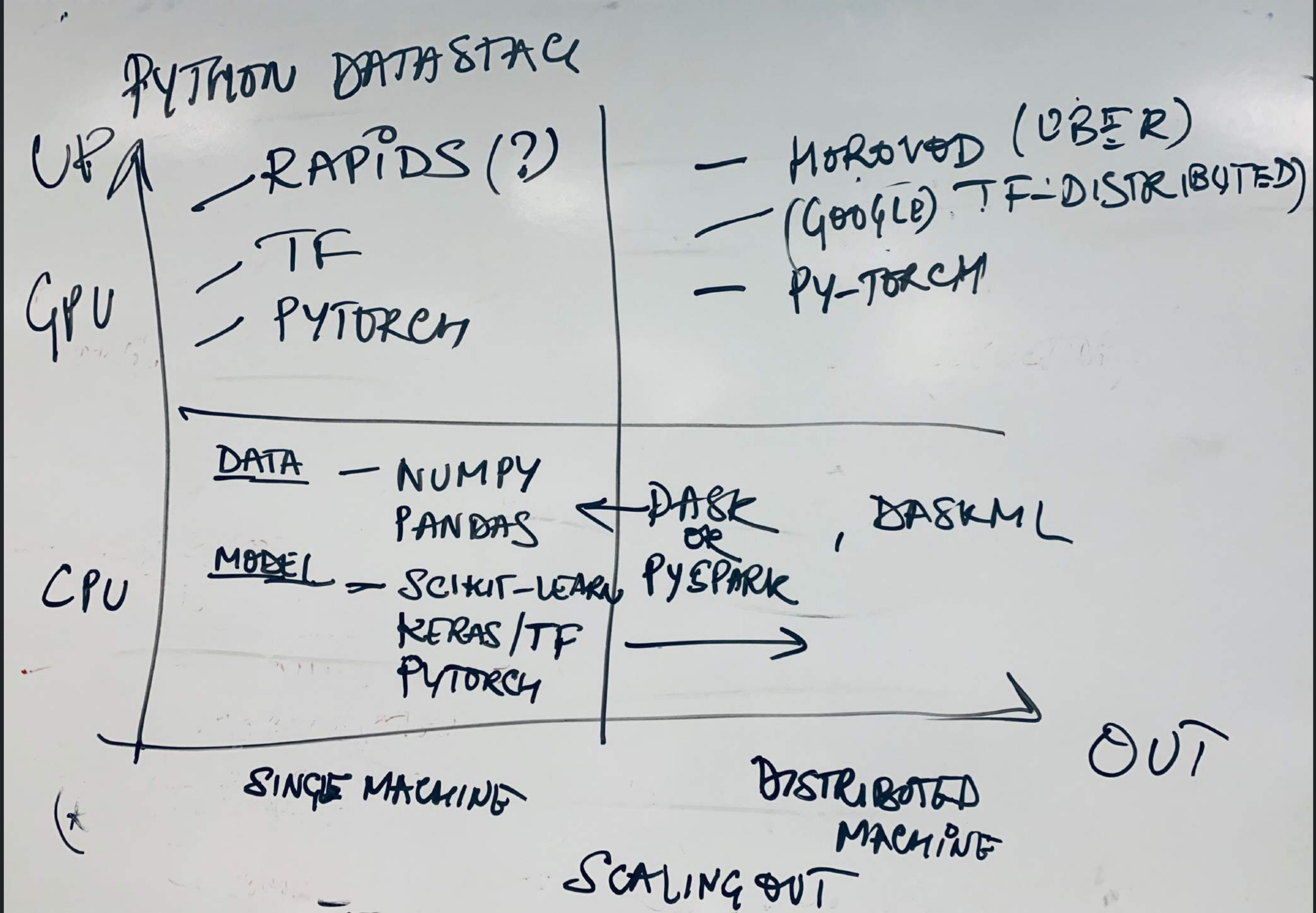
Yogi Berra

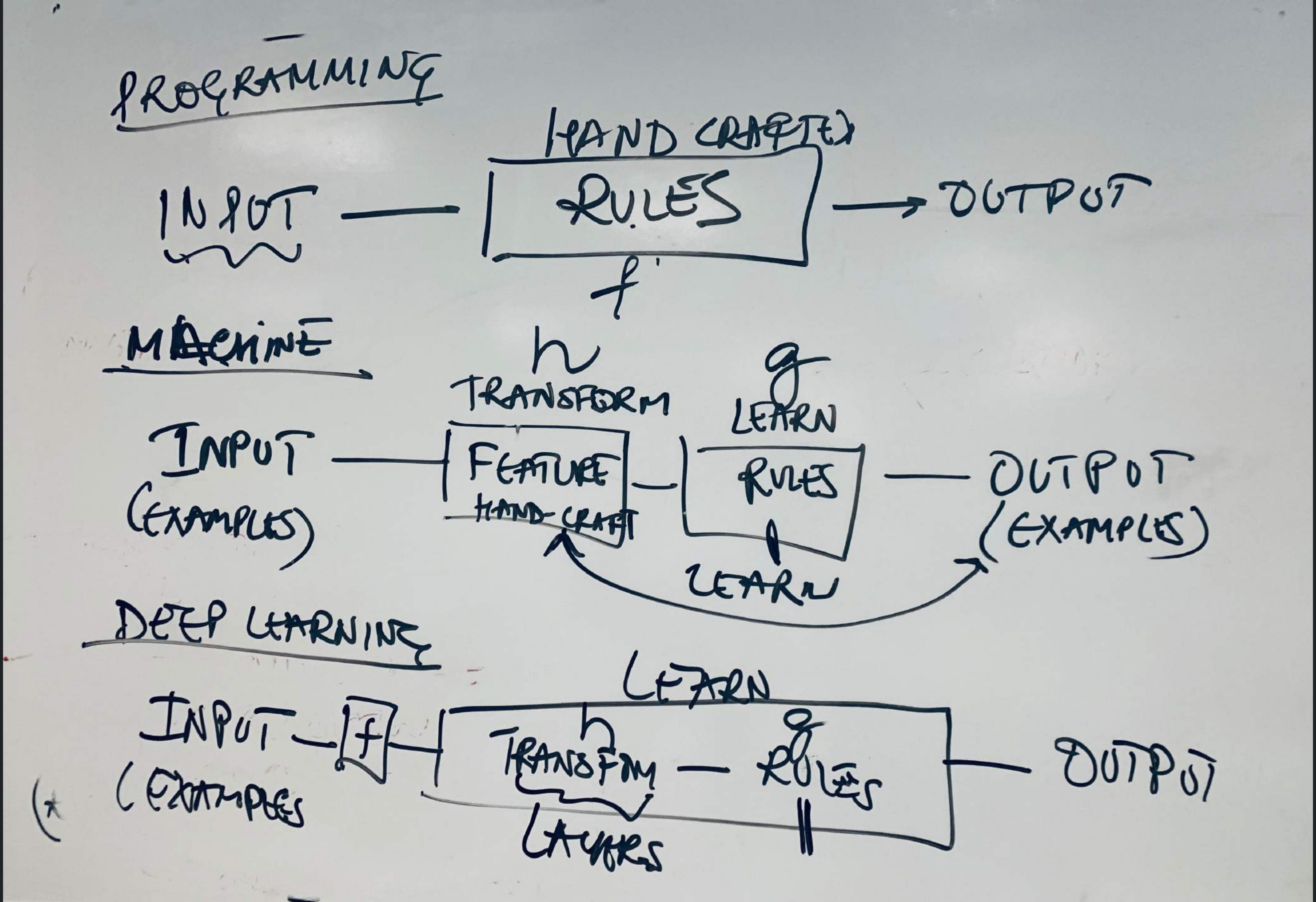
DL Collaborative Filtering

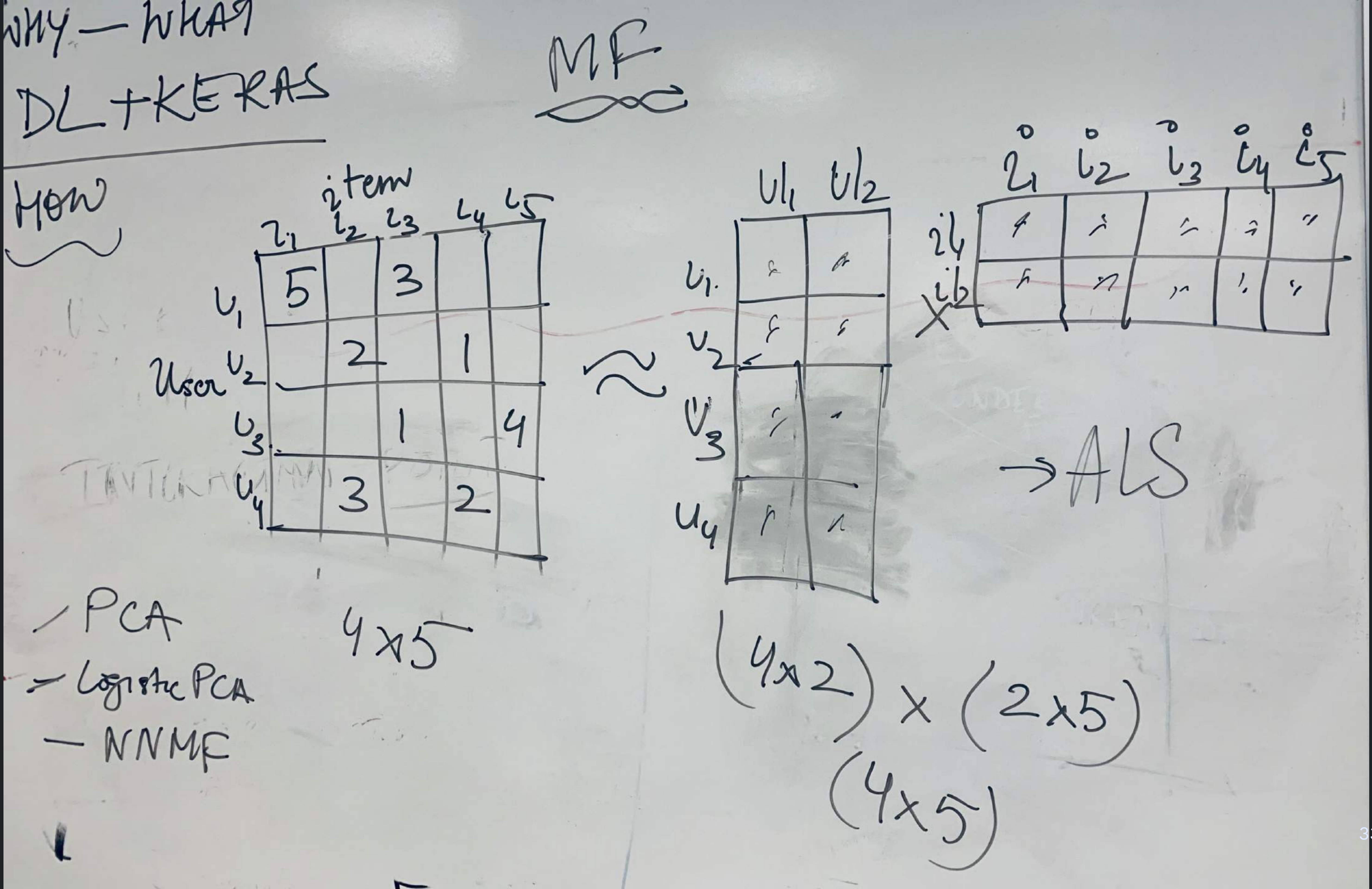
- Concepts: Python DS, DL basics, Matrix Factorisation & Embeddings
- Build: Five models with DL
  - Matrix Factorisation
  - Matrix Factorisation with Bias
  - Non-Negative Matrix Factorisation
  - Deep Matrix Factorisation
  - Neural Collaborative Filtering

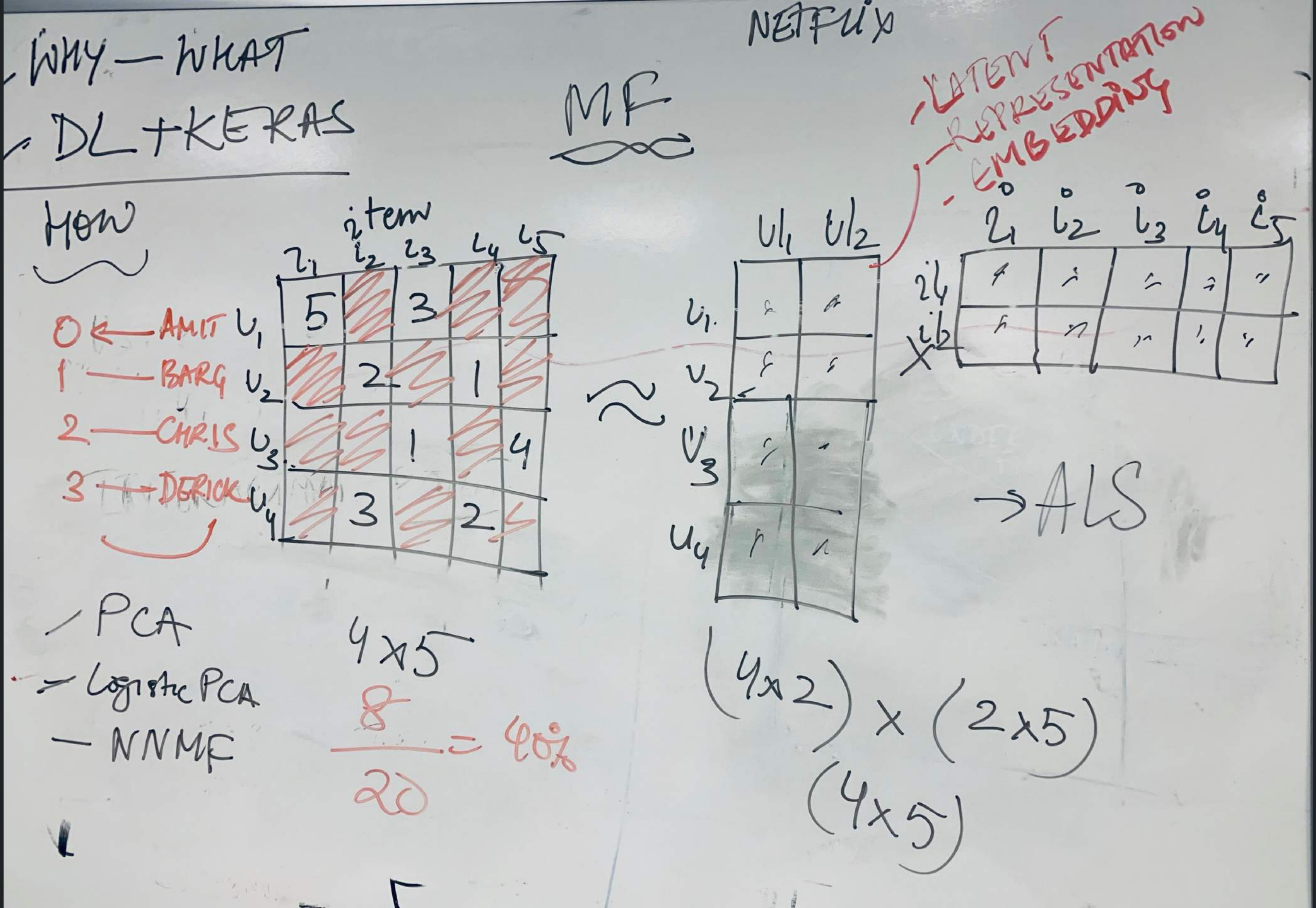
Data Preparation

- Data Preprocessing
  - Index Users (Label Encoding)
  - Index Items (Label Encoding)
  - Ratings: Explicit or Implict
- Data Splitting
  - Random
  - StratifiedL User or Item based
  - Chronological: time-based


PYTHON DATASTACK
UP
GPU
CPU
RAPIDS (?)
TF
PYTORCH
HOROVOD (UBER)
(GOOGLE) TF-DISTRIBUTED)
PY-TORCH
DATA
NUMPY
PANDAS
DASK
OR
PYSPARK
, DASKML
MODEL
SCIKIT-LEARN
KERAS/TF
PYTORCH
SINGLE MACHINE
DISTRIBUTED
MACHINE
OUT
SCALING OUT


PROGRAMMING
HAND CRAFTED
INPUT
RULES
OUTPUT
f
MACHINE
h
TRANSFORM
g
LEARN
INPUT
(EXAMPLES)
FEATURE
HAND-CRAFT
RULES
OUTPUT
(EXAMPLES)
LEARN
DEEP LEARNING
LEARN
INPUT
(EXAMPLES
f
h
TRANSFORM
g
RULES
LAYERS
OUTPUT


WHY — WHAT
DL+KERAS
MF
HOW
item
User
5
3
2
1
1
4
3
2
4x5
PCA
Logistic PCA
NNMF
(4x2) x (2x5)
(4x5)
→ ALS



- WHY — WHAT
- DL + KERAS
NETFLIX
MF
- LATENT
- REPRESENTATION
- EMBEDDING
HOW
item
0 ← AMIT
1 — BARG
2 — CHRIS
3 — DERICK
- PCA
- Logistic PCA
- NNMF
4x5
8/20 = 40%
(4x2) x (2x5)
(4x5)
→ ALS

Content Based Recommenders

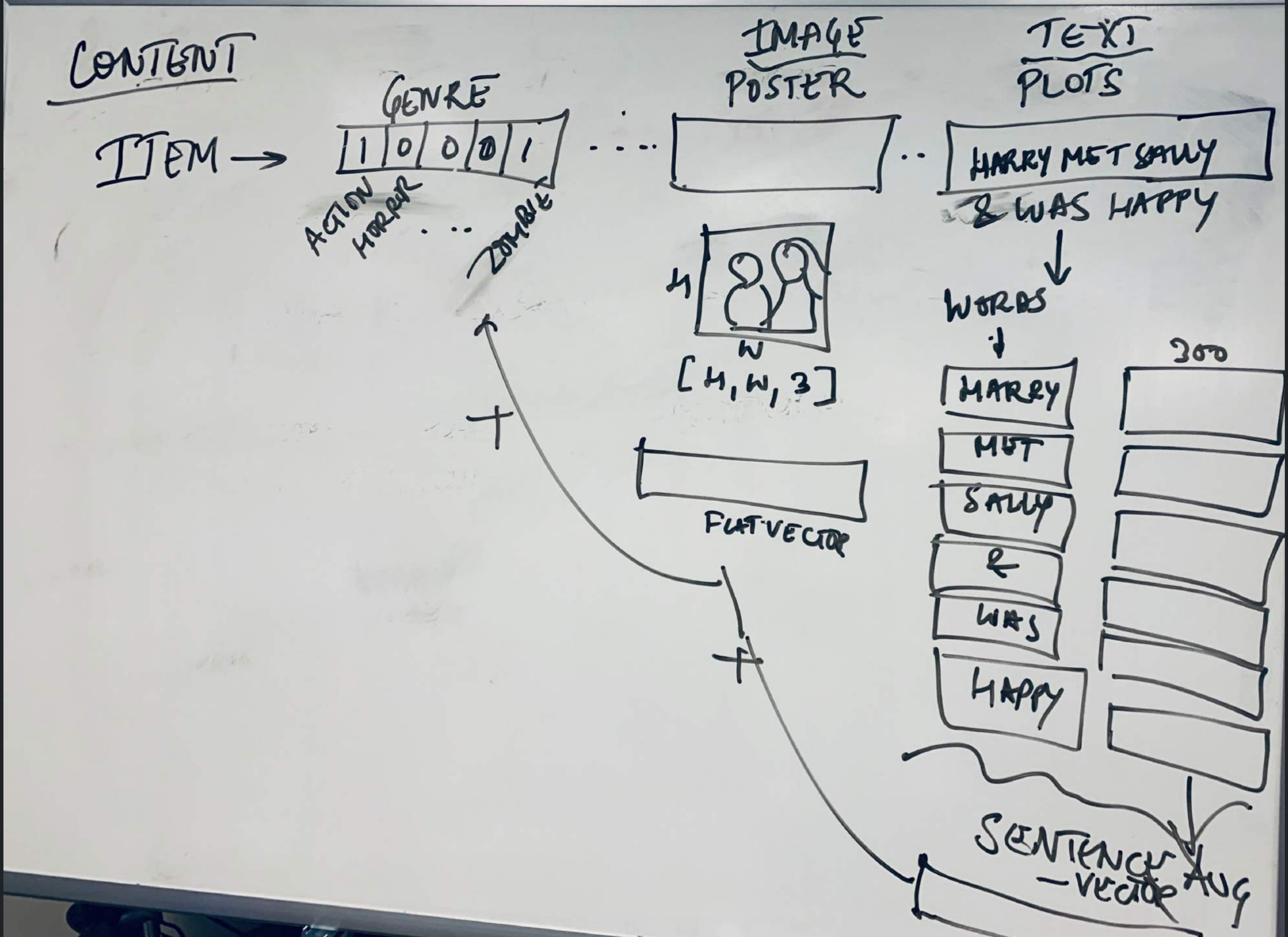
- **Concept – Embedding:** Learning the representation for items
  - Exemplar: Pandora: 400 features for each song
  - Item Metadata: Continuous, Categorical, Text , Video, Image, Sound => Heterogeneous data
  - Distance Metrics: Dot product, Cosine, Euclidean
- **Build**: Three models for evaluation
  - Baseline: Popularity Based
  - Categorical + Sparse Embedding (Text)
  - Categorical + Dense Embedding (Pre-trained W2V)

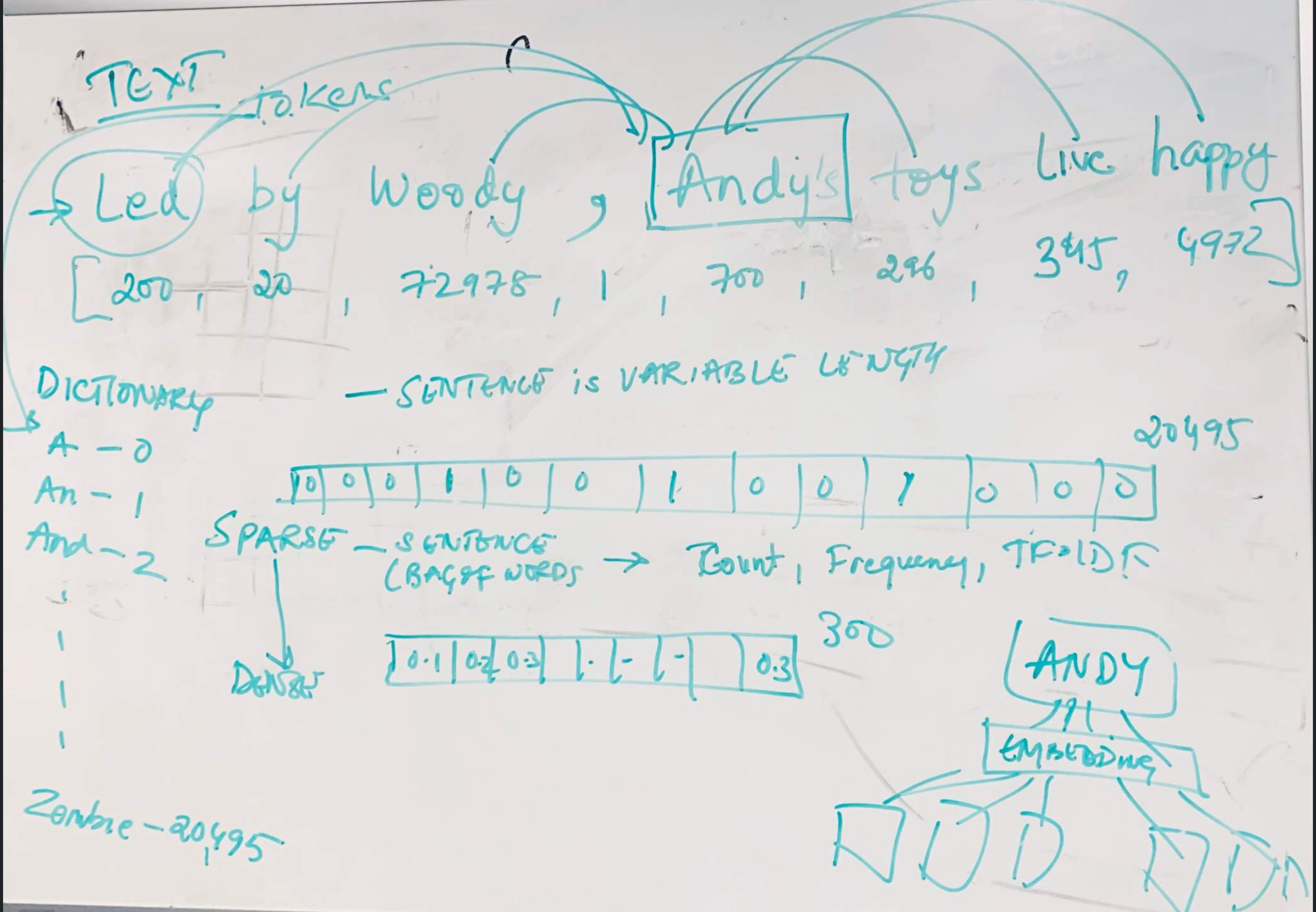
- FEATURE EXTRACTION
  - CONTENT + CF

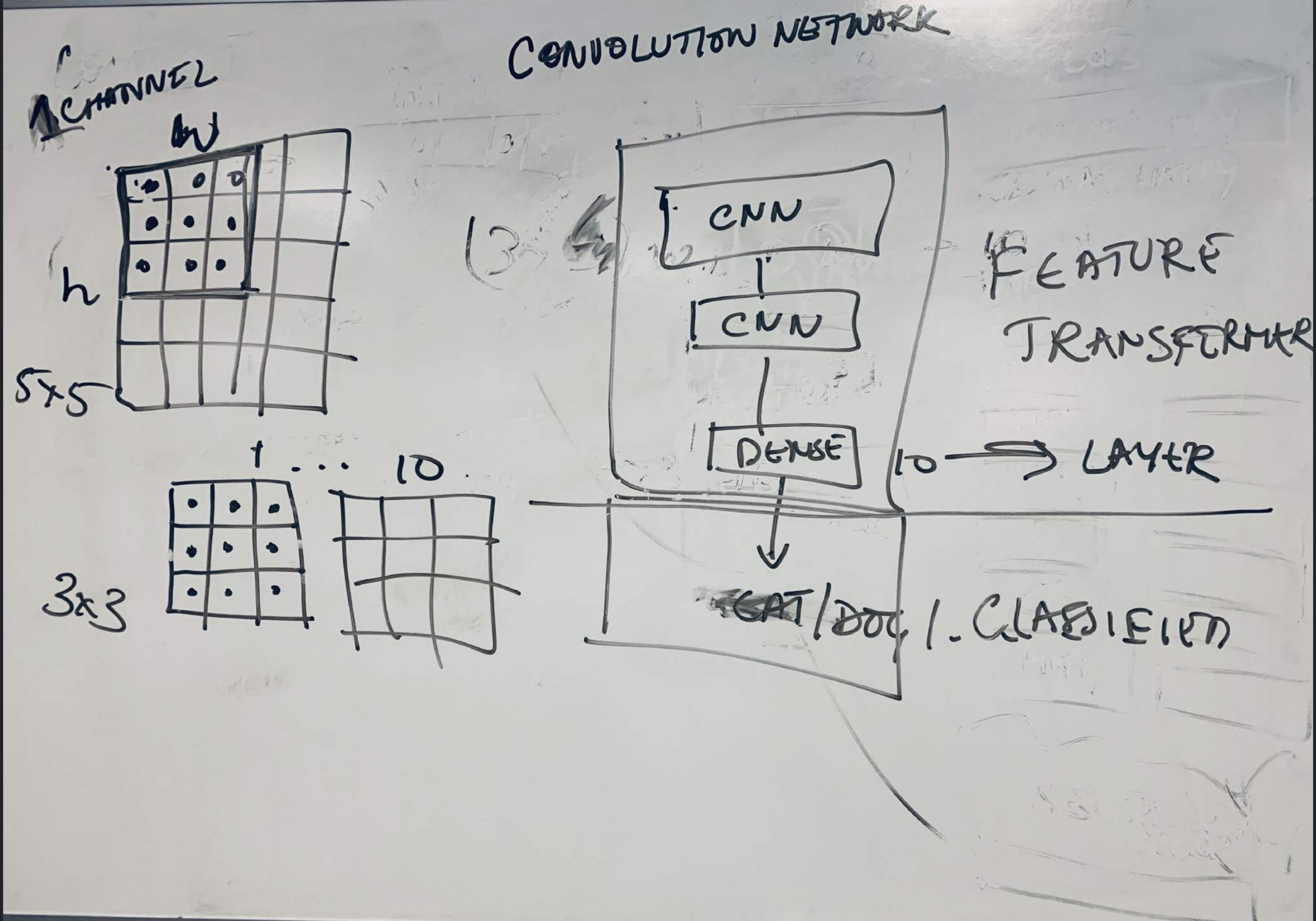
- RANKING
- REAL DATA : Session → MODEL

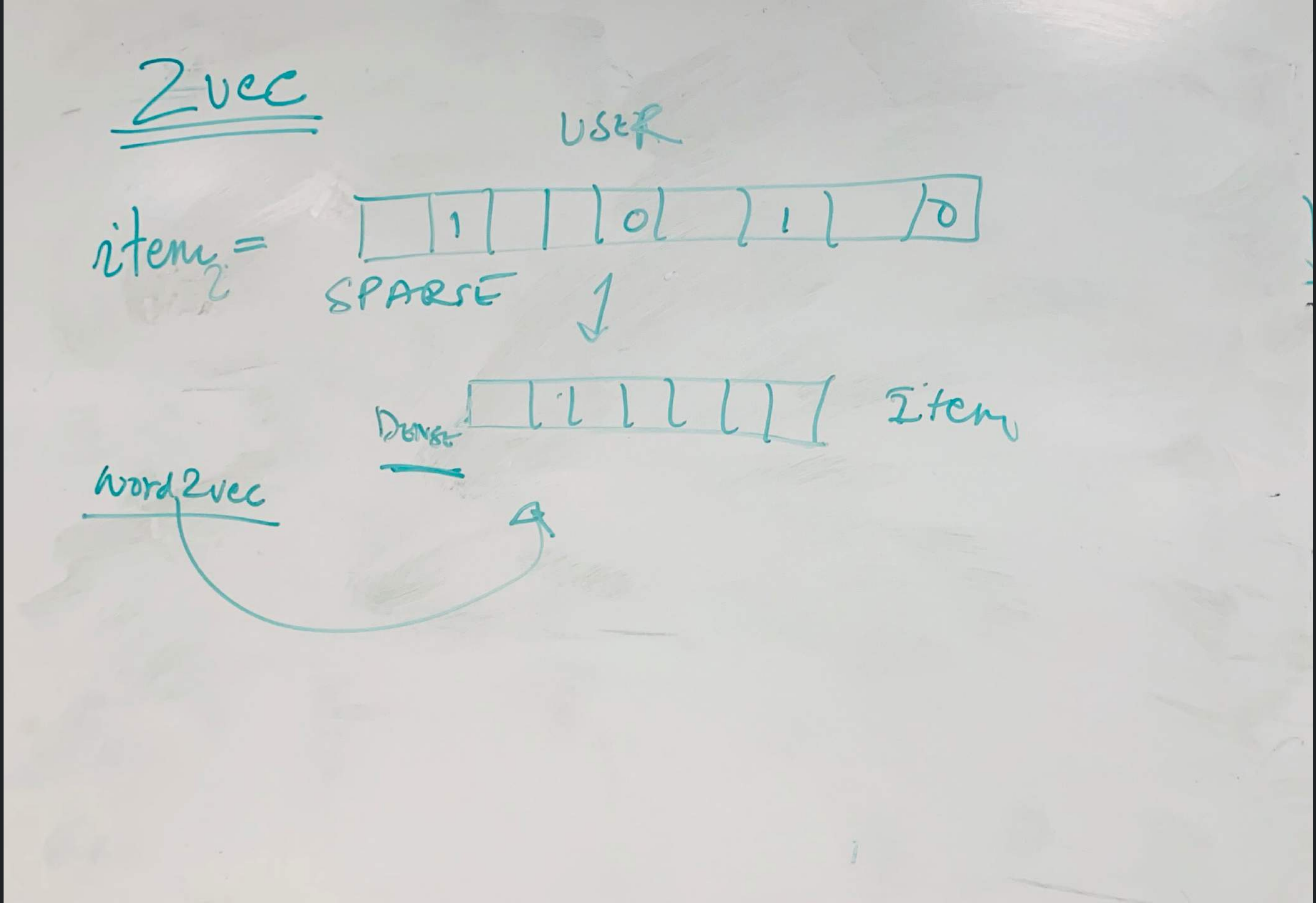
IMPLICIT

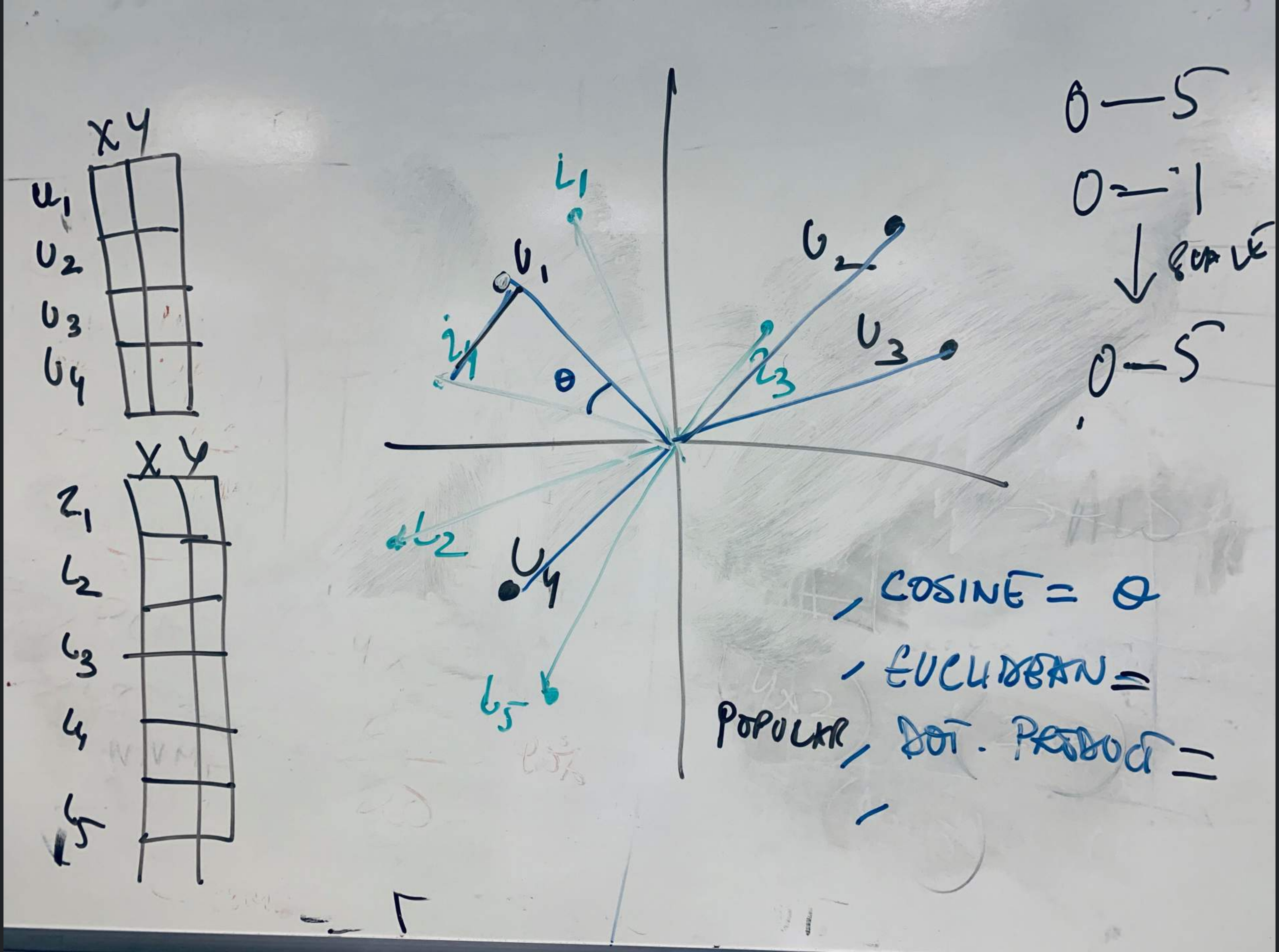
→ DEPLOYMENT, SCALABILITY LONGITITUDINAL
EXPLAINABILITY


CONTENT
ITEM
GENRE
1 0 0 0 1
ACTION
HORROR
ZOMBIE
IMAGE
POSTER
H
W
[H, W, 3]
FLAT VECTOR
TEXT
PLOTS
HARRY MET SALLY & WAS HAPPY
WORDS
HARRY
MET
SALLY
&
WAS
HAPPY
300
SENTENCE
-VECTOR
AVG


TEXT
TOKENS
Led by Woody, Andy's toys live happy
[200, 20, 72978, 1, 700, 296, 3415, 4972]
DICTIONARY
A - 0
An - 1
And - 2
Zombie - 20,495
— SENTENCE is VARIABLE LENGTH
20495
0 0 0 1 0 0 1 0 0 1 0 0 0
SPARSE — SENTENCE (BAG OF WORDS) → Count, Frequency, TF-IDF
300
DENSE
0.1 0.2 0.3 0.3
ANDY
EMBEDDING


CONVOLUTION NETWORK
1 CHANNEL
w
h
5x5
1 ... 10
3x3
CNN
CNN
DENSE
FEATURE
TRANSFORMER
10 → LAYER
CAT/DOG
1. CLASSIFIER

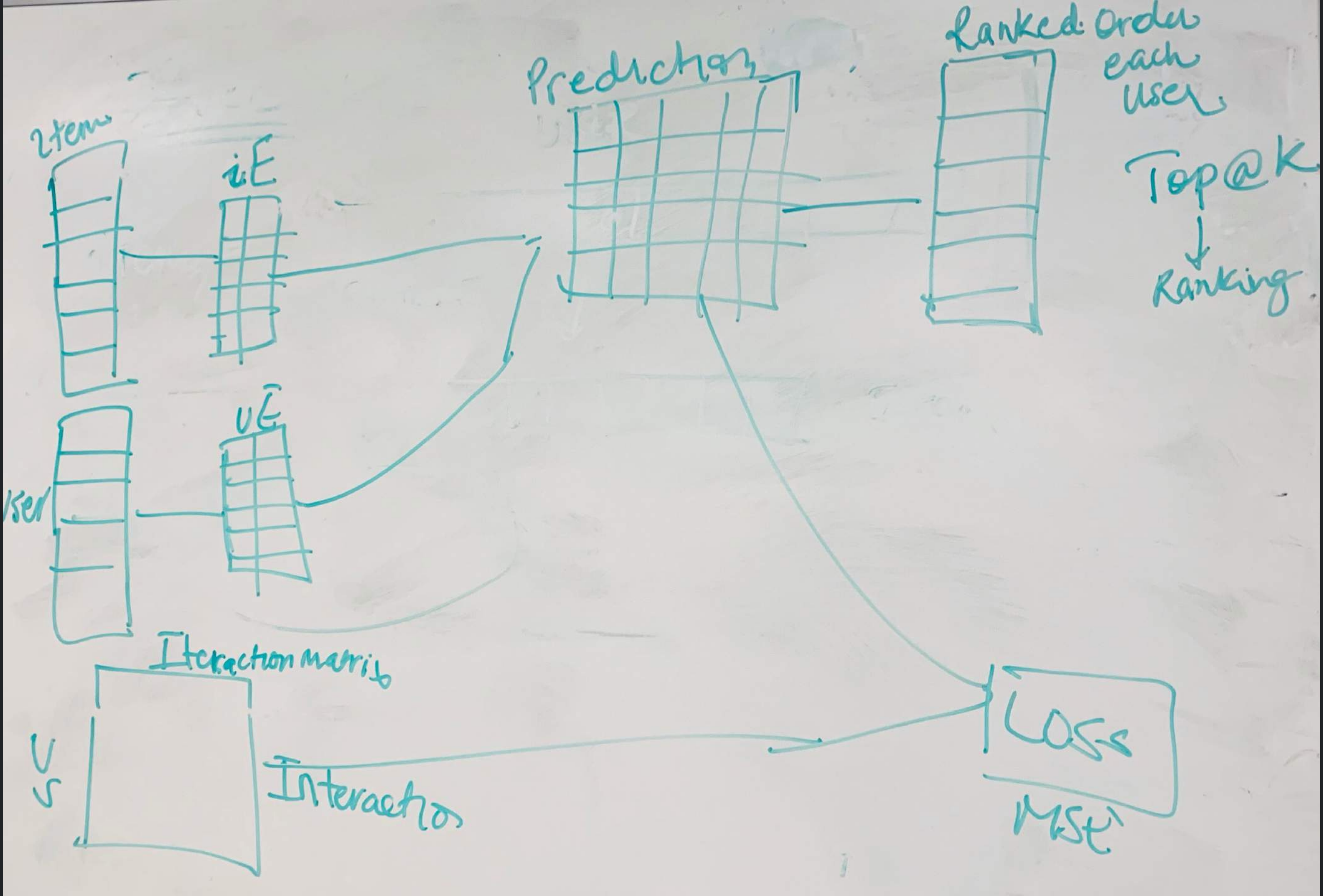

2vec
USER
item2 =
1
0
1
0
SPARSE
Dense
Item
word2vec

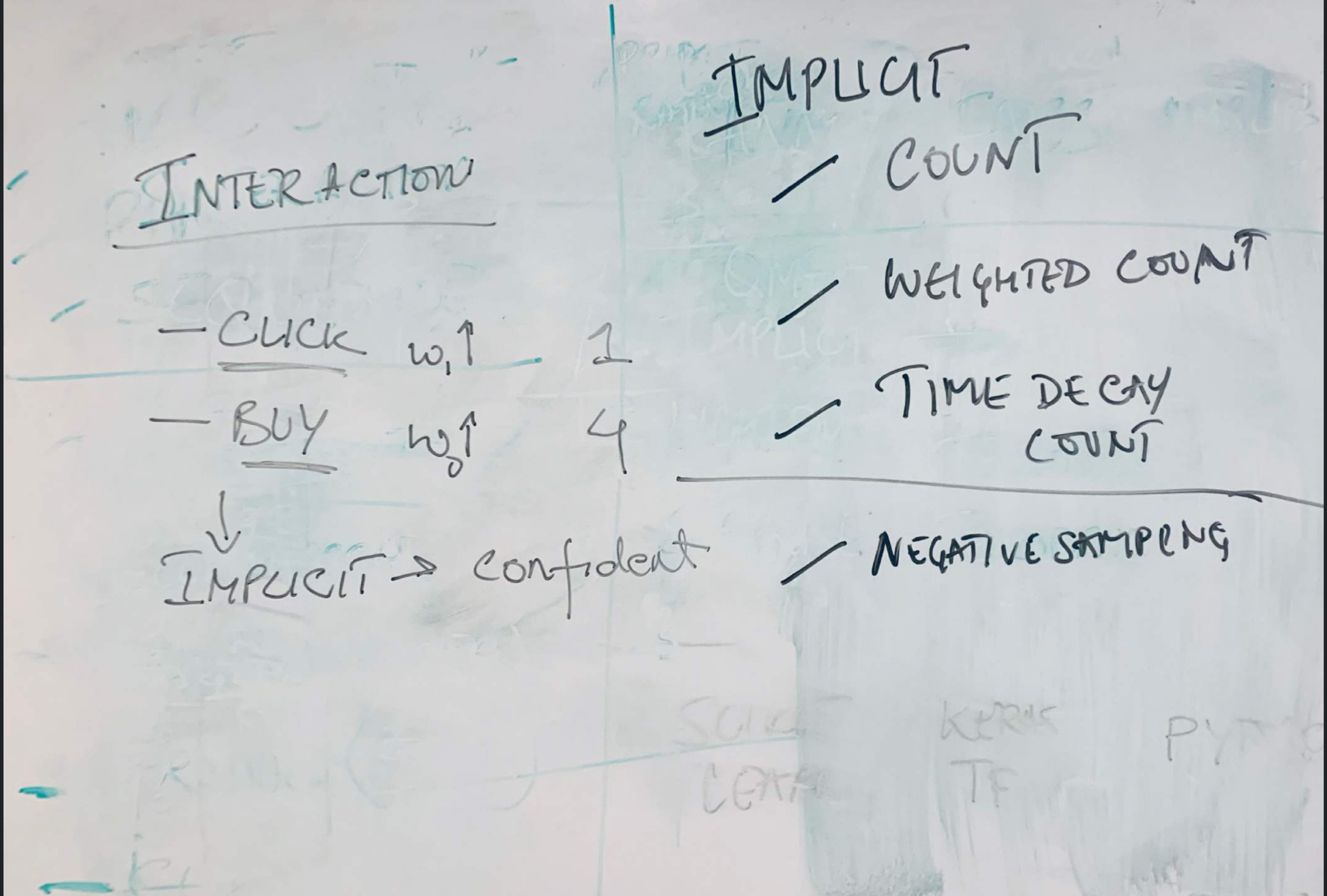


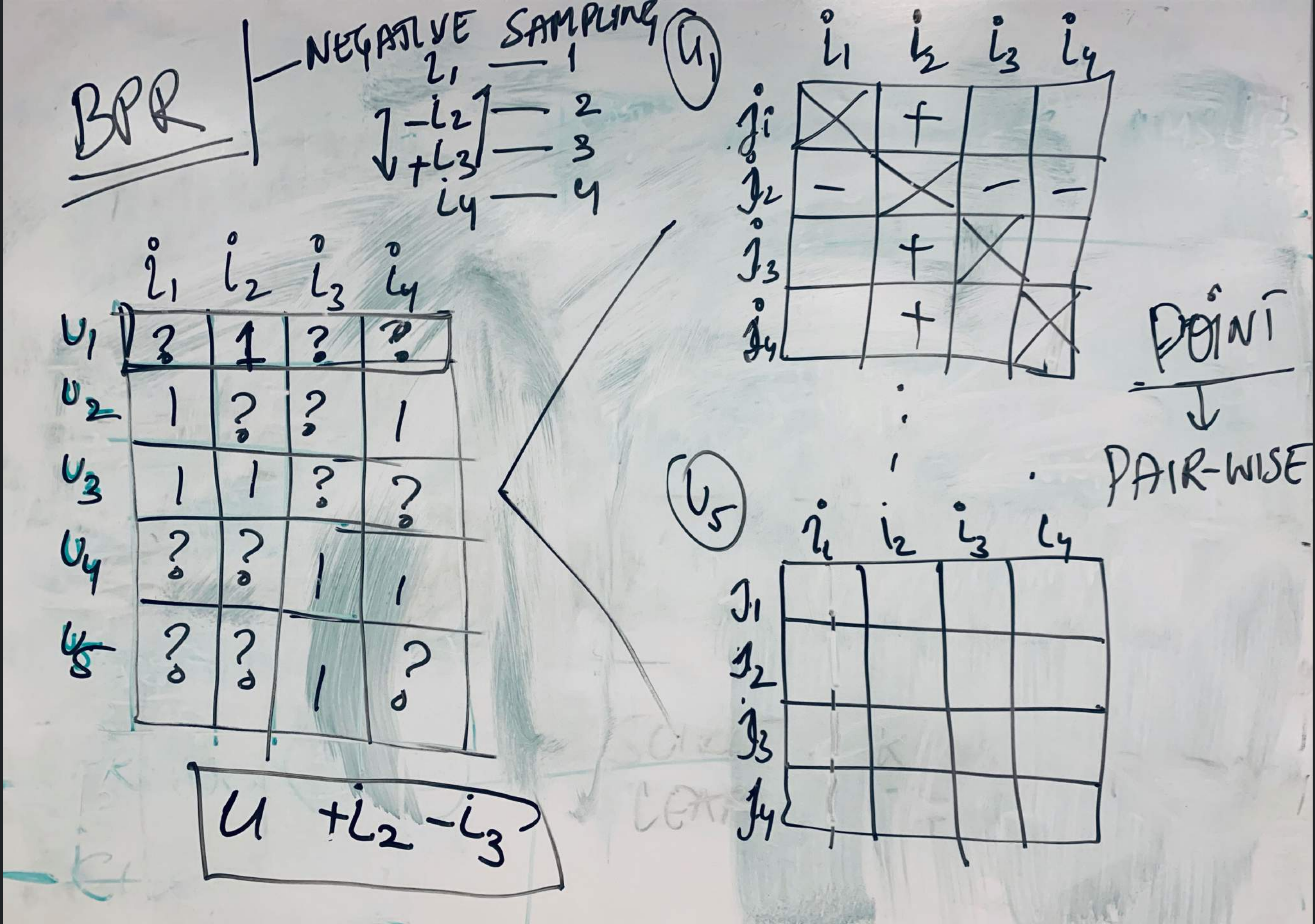
X Y
u1
u2
u3
u4
X Y
i1
i2
i3
i4
i5
i1
U1
i1
θ
U2
U3
i3
i2
U4
i5
0 — 5
0 — 1
SCALE
0 — 5
COSINE = θ
EUCLIDEAN =
POPULAR
DOT . PRODUCT =

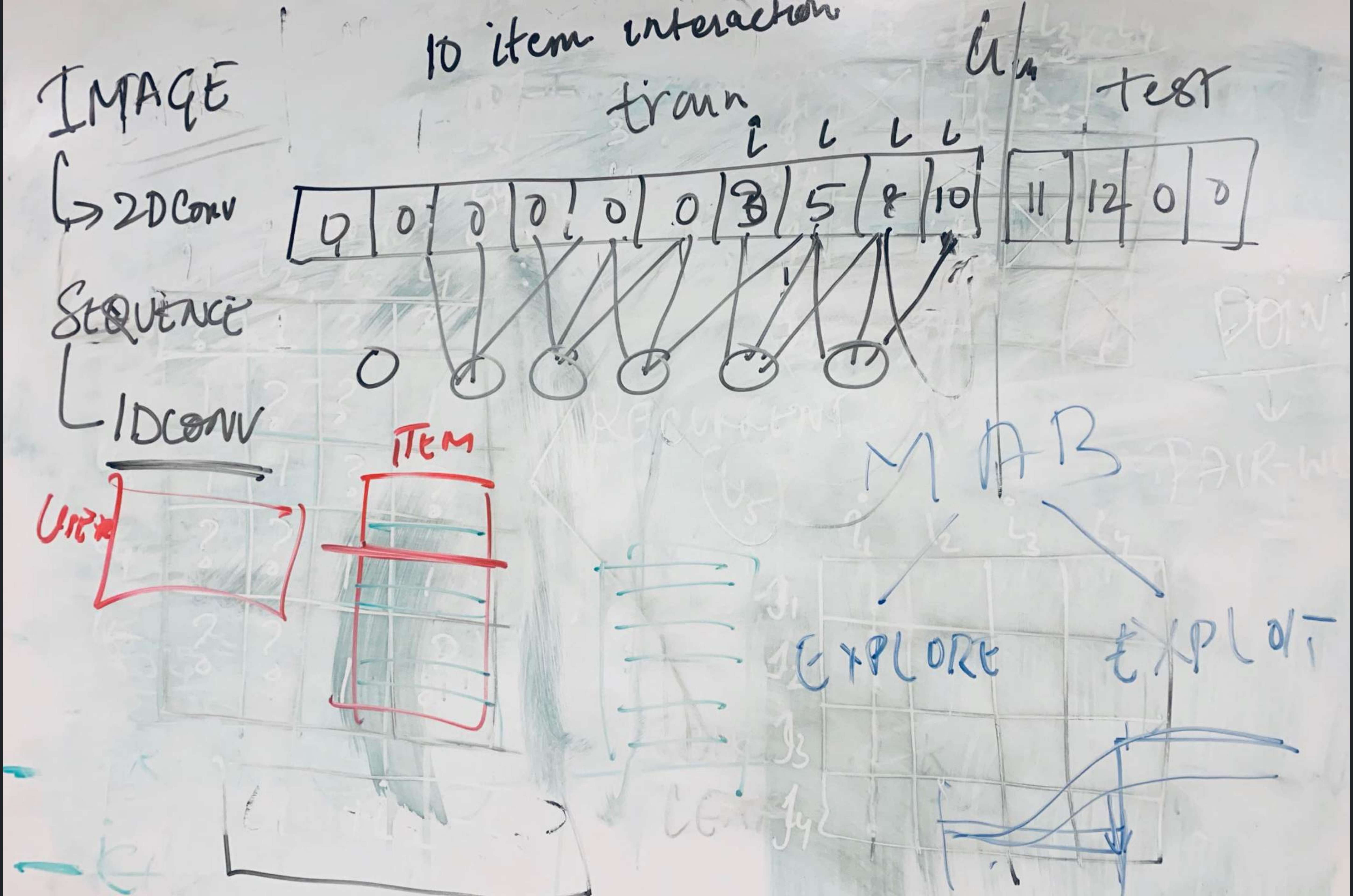
Learning to Rank

- Ranking vs. Rating
- Candidate Generation -> Ranking -> Re-ranking
- Complexity (High number of combinations)
- Point vs. Pair vs Triplet vs. List
- Loss: BPR, WARP
- Evaluation: MAP@k, NDCG@k
- Models:
  - Point Based (Explicit vs. Implicit)
  - Sequence Based

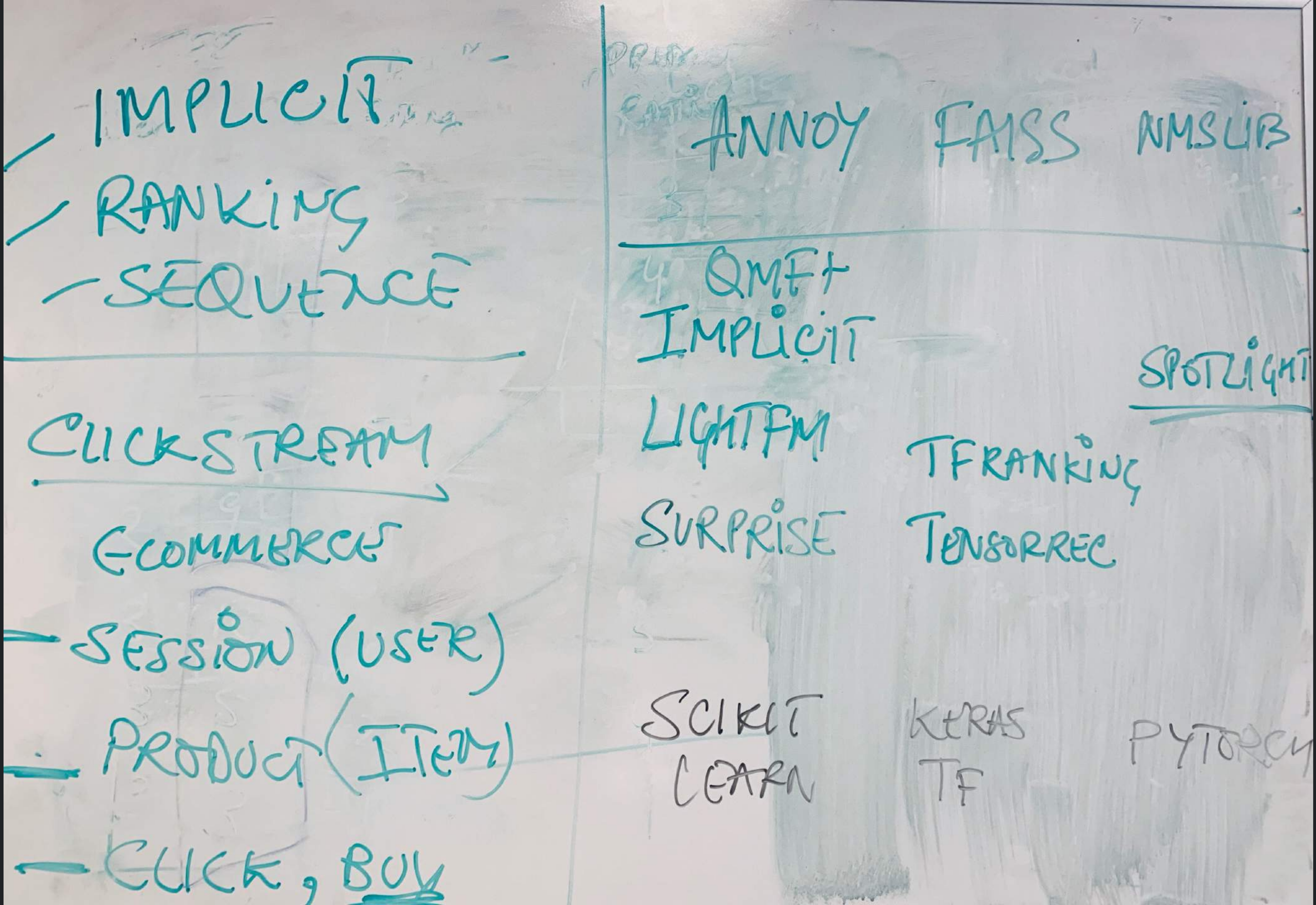

Item
iE
User
uE
Prediction
Ranked Order each user
Top@K
Ranking
Interaction matrix
U
I
Interaction
Loss
MSE


INTERACTION
— CLICK $w_1 \uparrow$ 1
— BUY $w_0 \uparrow$ 4
IMPLICIT → confident
IMPLICIT
COUNT
WEIGHTED COUNT
TIME DECAY COUNT
NEGATIVE SAMPLING


BPR
NEGATIVE SAMPLING
i1 — 1
-i2 — 2
+i3 — 3
i4 — 4
U1
U5
POINT
PAIR-WISE
U +i2 -i3



10 item interaction
IMAGE
2DConv
SEQUENCE
1DConv
train
test
9 0 0 0 0 0 3 5 8 10
11 12 0 0
ITEM
MAB
EXPLORE
EXPLOIT


IMPLICIT
RANKING
SEQUENCE
CLICKSTREAM
ECOMMERCE
SESSION (USER)
PRODUCT (ITEM)
CLICK, BUY
ANNOY
FAISS
NMSLIB
QMF+
IMPLICIT
SPOTLIGHT
LIGHTFM
TFRANKING
SURPRISE
TENSORREC
SCIKIT
LEARN
KERAS
TF