

# Search and Recommendation

Hao Sheng  
August 10th, 2023

# Agenda

- **Introduction**
- **Lecturing:** Search Engine System
- **Break-out:** Recommendation system in daily life
- **Lecturing:** Advanced Topics (Part I)
- **Lab:** Recommendation system notebook II
- **Lecturing:** Advanced Topics (Part II)

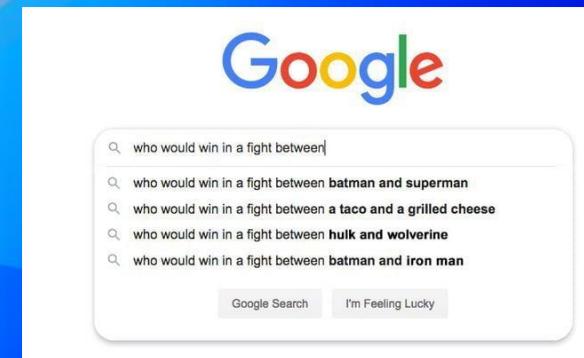


60% Lecturing

25% Lab

10% Discussion

# Search Engine System

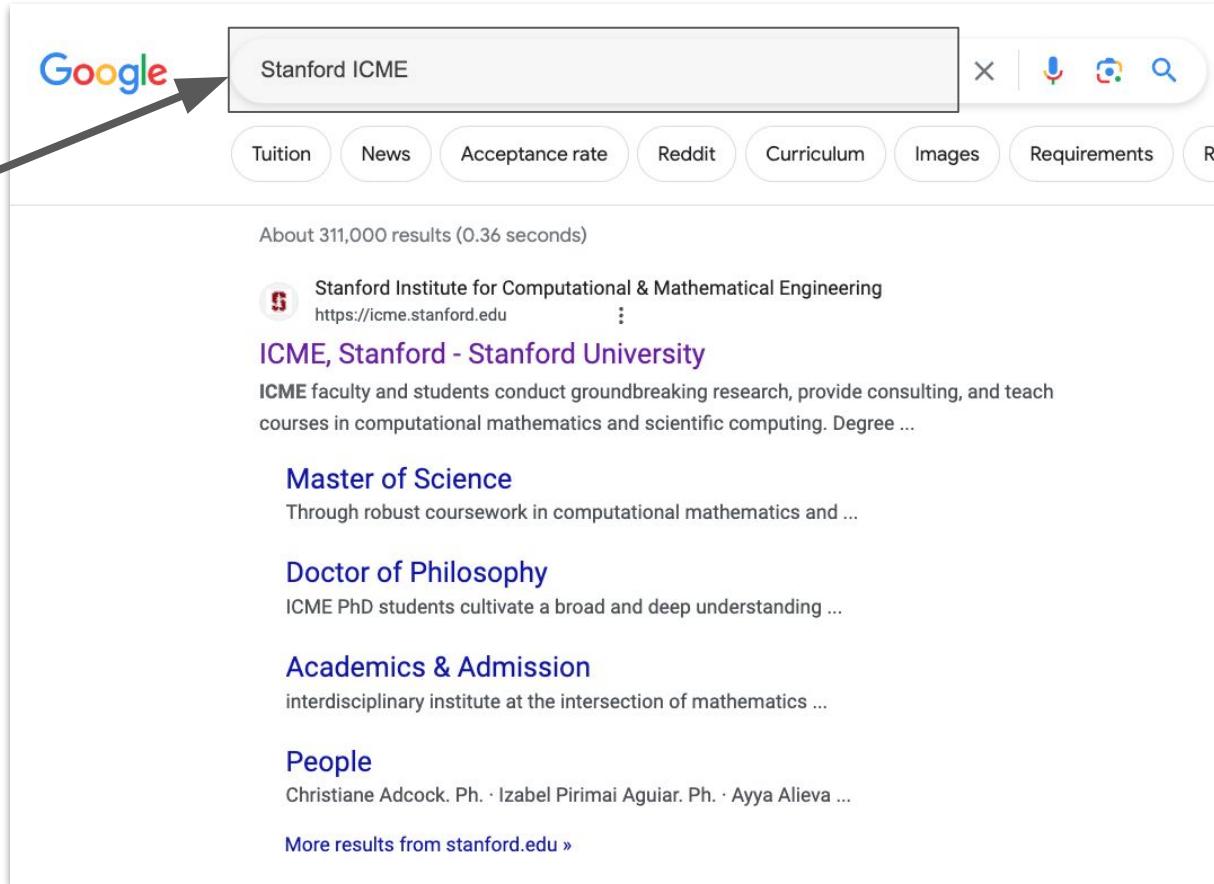


# Quick Recap: Movie Recommendation System



Sara	5	3		2	2	2
Jesper	4	3	4		3	3
Therese	5	2	5	2	1	1
Helle	3	5	3		1	1
Pietro	3	3	3	2	4	5
Ekaterina	2	3	2	3	5	5

Search Query



A screenshot of a Google search results page. The search bar at the top contains the query "Stanford ICME". Below the search bar are several navigation links: "Tuition", "News", "Acceptance rate", "Reddit", "Curriculum", "Images", "Requirements", and "Results". A large arrow points from the text "Search Query" on the left towards the search bar. The search results section shows the following information:

About 311,000 results (0.36 seconds)

 Stanford Institute for Computational & Mathematical Engineering  
<https://icme.stanford.edu> ::

**ICME, Stanford - Stanford University**  
ICME faculty and students conduct groundbreaking research, provide consulting, and teach courses in computational mathematics and scientific computing. Degree ...

**Master of Science**  
Through robust coursework in computational mathematics and ...

**Doctor of Philosophy**  
ICME PhD students cultivate a broad and deep understanding ...

**Academics & Admission**  
interdisciplinary institute at the intersection of mathematics ...

**People**  
Christiane Adcock. Ph. · Izabel Pirimai Aguiar. Ph. · Ayya Alieva ...

[More results from stanford.edu »](#)

# Search Engine: First Glance

The diagram illustrates a search engine results page for the query "Institute for Computational and Mathematical Engineering". It features four search results cards arranged vertically, each with a small icon, the source name, a URL, and a truncated description. Four arrows point from the right side of the slide towards these cards.

- Stanford University**  
https://events.stanford.edu › department › institute\_for...  
**Institute for Computational and Mathematical Engineering ...**  
The Institute for Computational & Mathematical Engineering (ICME) is a degree granting (M.S./Ph.D.) interdisciplinary institute at the intersection of ...
- Stanford University Bulletin**  
https://bulletin.stanford.edu › programs › CME-MS ...  
**CME-MS Program - Stanford Bulletin**  
ICME is a degree granting (M.S./Ph.D.) interdisciplinary institute at the intersection of mathematics, computing, engineering and applied sciences.  
Courseor course: Intermediate Econometrics II (3 ...
- Stanford Bulletin Archive**  
https://archived-bulletin.stanford.mobi › institutefor...  
**Institute for Computational and Mathematical Engineering**  
At ICME, we design state-of-the-art mathematical and computational models, methods, and algorithms for engineering and science applications. The program ...  
Or MS&E 327: Topics in Causal Inference      STATS 263: Design of Experiments
- LinkedIn**  
https://www.linkedin.com › company › icme-stanford ...  
**Institute for Computational and Mathematical Engineering ...**  
Institute for Computational and Mathematical Engineering at Stanford University (ICME) | 1056 followers on LinkedIn. Groundbreaking research into complex ...

Search Engine Results  
(Websites)

# Search Engine: First Glance

Stanford Online  
<https://online.stanford.edu/programs/computationa...>

## Computational and Mathematical Engineering MS Degree

The Institute for Computational and Mathematical Engineering (ICME) is a degree granting institute at the intersection of mathematics, computing, engineering ...

<https://twitter.com/ICMESTanford>

### Stanford ICME (@ICMESTanford) · Twitter

Generative Models (SWS 14)  
Aug. 9-10  
1-4pm PDT  
**ICME**  
Stanford Computational  
Explore methods for enhancing #UX & retrieving information in @Stanford ICME's new Search and Recommendation workshop from 8/9 to 8/10, led by @Apple Staff Machine Learning Engineer @hao\_ss.

Search and Recommendation (SWS 13)  
Aug. 9-10  
8-11am PDT  
**ICME**  
Stanford Computational  
Explore methods for enhancing #UX & retrieving information in @Stanford ICME's new Search and Recommendation workshop from 8/9 to 8/10, led by @Apple Staff Machine Learning Engineer @hao\_ss.

Introduction to Natural Language Processing (SWS 12)  
Aug. 7-8  
1-4pm PDT  
**ICME**  
Stanford Computational  
Learn the building blocks of modern #NLP concepts in @Stanford ICME's Intro course from 8/7 to 8/8, taught by brothers and @Google Software Engineers @afshinea & @shervinea.

Register:  
[www.eventbrite.com/e/ic...](http://www.eventbrite.com/e/ic...)

Twitter · Jul 28, 2023

Twitter · Jul 27, 2023

Twitter · Jul 26, 2023

Search Engine Results  
(Tweets)



Stanford University  
[https://events.stanford.edu/department/institute\\_for...](https://events.stanford.edu/department/institute_for...)

## Institute for Computational and Mathematical Engineering ...

The Institute for Computational & Mathematical Engineering (ICME) is a degree granting (M.S./Ph.D.) interdisciplinary institute at the intersection of ...

# Search Engine: First Glance

People also ask :

- How competitive is Stanford graduate school? ▾
- What is the Toefl code for Stanford ICME? ▾
- Is Stanford University good for engineering? ▾
- What is the mathematical and computational finance program at Stanford University? ▾

Feedback

 Stanford Online  
<https://online.stanford.edu> › programs › computationa... ::

## Computational and Mathematical Engineering MS Degree

The Institute for Computational and Mathematical Engineering (ICME) is a degree granting institute at the intersection of mathematics, computing, engineering ...



Search Engine Results  
(Tweets)

# Search Engine v.s. Movie Recommendation

- It has a search bar!
- The items are mal-defined at the first glance.
- User does not simply rate the search results!



“Recommender systems (RSs) are software tools and techniques that provide suggestions for items that are most likely of interest to a particular user. ”

--- *Introduction to Recommender Systems Handbook*

# Search Engine as a Recommendation System

“Recommender systems (RSs) are software tools and techniques that provide suggestions for items that are most likely of interest to a particular user. ”

--- *Introduction to Recommender Systems Handbook*

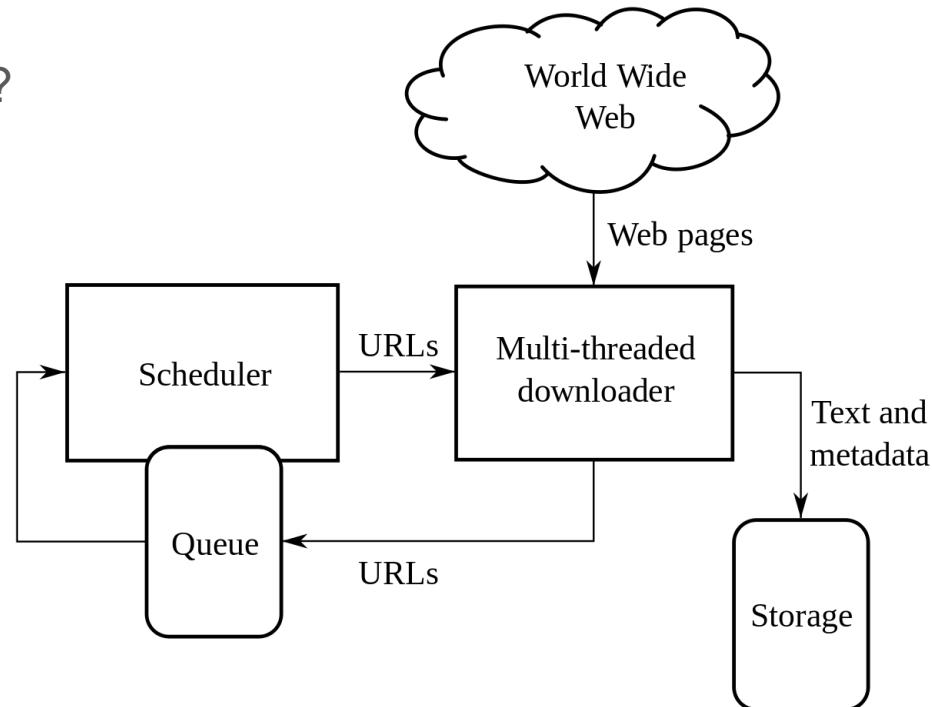
Search Engine is a **Recommendation System!**

# Search Engine: Learning Goals

- It has a search bar!
  - -> How to incorporate the user intention?
- The items are mal-defined at the first glance.
  - -> How to crawl the internet and store the items?
- User does not simply rate the search results!
  - -> How to assign user-item rating with user data?

Where to get the search results (items)?

The internet!



Where to get the search results (items)?

The internet!

```
class Spider:
    name = 'icme_spider'
    start_urls = 'https://icme.stanford.edu/'
    parsed_urls = []

    def parse(self, url: str):
        self.parse_url.append(url)
        for next_url in Website(url):
            self.parse(next_url)
```

# Search Engine: Web Crawler - Recursion

```
parsed_urls = ["https://icme.stanford.edu"]
```

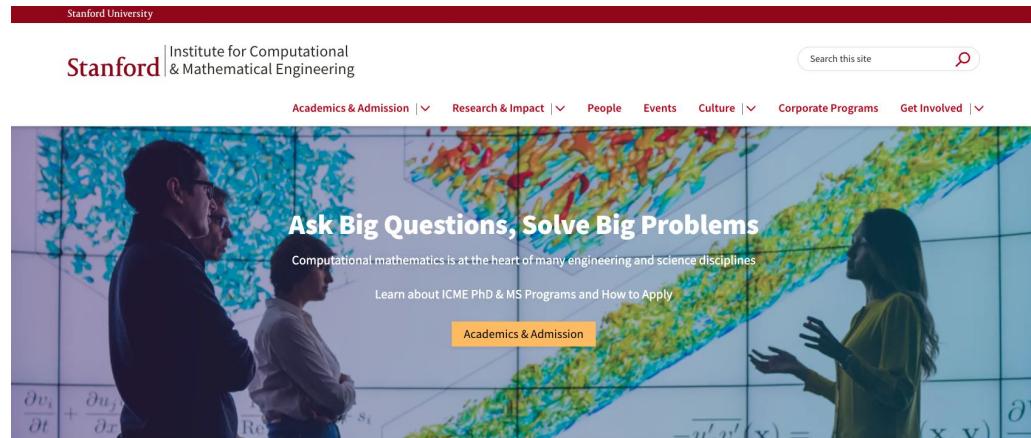
```
class Spider:
    name = 'icme_spider'
    start_urls = 'https://icme.stanford.edu/'
    parsed_urls = []

    def parse(self, url: str):
        self.parse_url.append(url)
        for next_url in Website(url):
            self.parse(next_url)
```

# Search Engine: Web Crawler - Recursion

```
parsed_urls = ["https://icme.stanford.edu"]
```

```
class Spider:  
    name = 'icme_spider'  
    start_urls = 'https://icme.stanford.edu/'  
    parsed_urls = []  
  
    def parse(self, url: str):  
        self.parse_url.append(url)  
        for next_url in Website(url):  
            self.parse(next_url)
```



## Events & Seminars

JUL

24

- to -

Workshop

[ICME Summer Workshops 2023 | Fundamentals of Data Science](#) ↗

April 10, 2023

[Counting Cars: New AI-Driven Approach Finds Tunnels Tell Tales](#) ↗

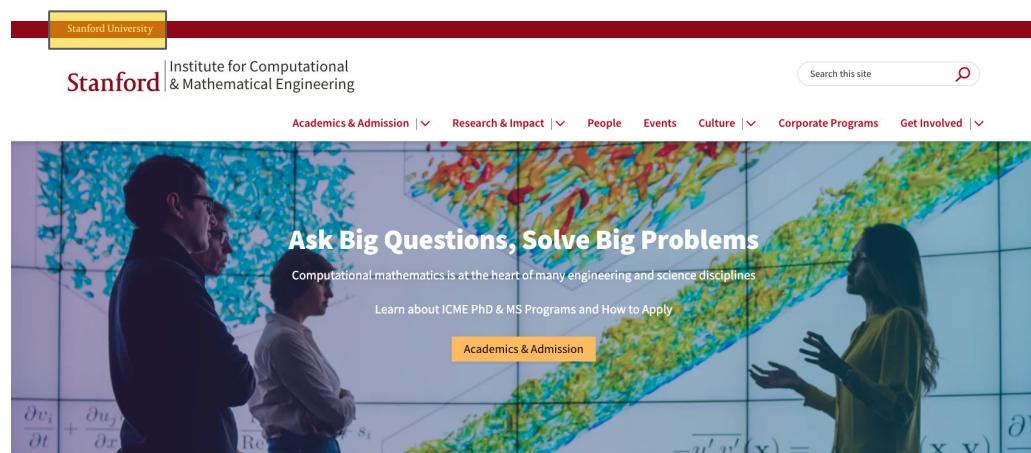
## News



# Search Engine: Web Crawler - Recursion

```
parsed_urls = ["https://icme.stanford.edu"]
```

```
class Spider:  
    name = 'icme_spider'  
    start_urls = 'https://icme.stanford.edu/'  
    parsed_urls = []  
  
    def parse(self, url: str):  
        self.parse_url.append(url)  
        for next_url in Website(url):  
            self.parse(next_url)
```



## Events & Seminars

JUL

24

- to -

Workshop

[ICME Summer Workshops 2023 | Fundamentals of Data Science](#) ↗

April 10, 2023

[Counting Cars: New AI-Driven Approach Finds Tunnels Tell Tales](#) ↗

## News

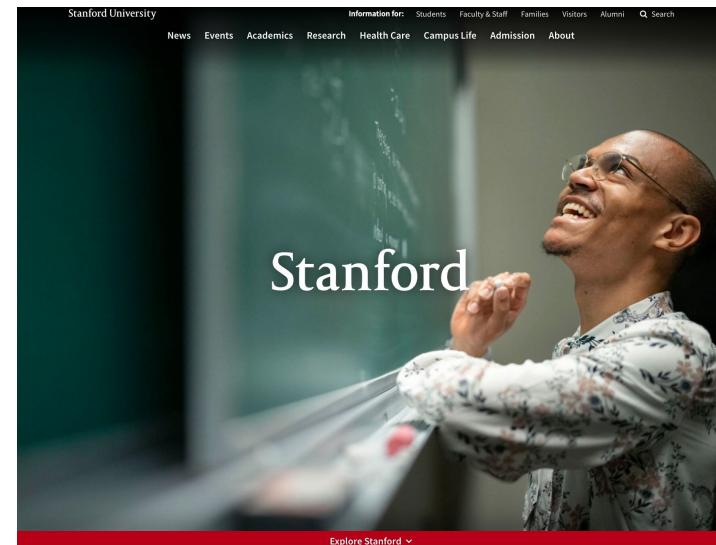


# Search Engine: Web Crawler - Recursion

parsed\_urls = ["https://icme.stanford.edu"]

```
class Spider:
    name = 'icme_spider'
    start_urls = 'https://icme.stanford.edu/'
    parsed_urls = []

    def parse(self, url: str):
        self.parse_url.append(url)
        for next_url in Website(url):
            self.parse(next_url)
```

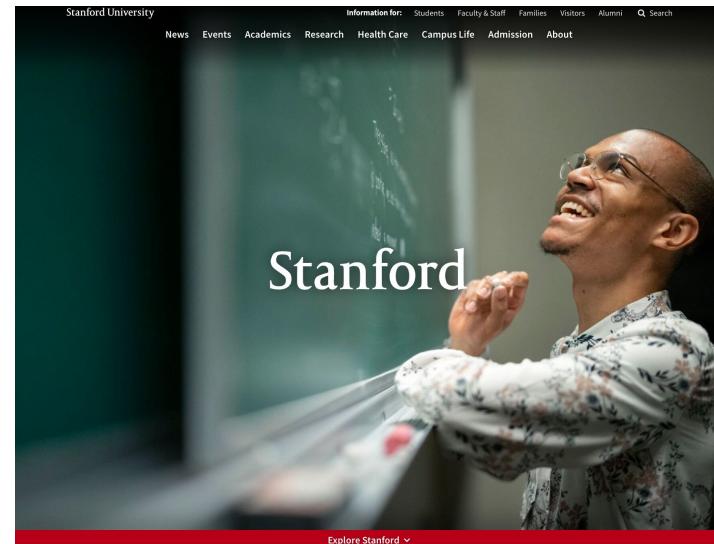


# Search Engine: Web Crawler - Recursion

```
class Spider:
    name = 'icme_spider'
    start_urls = 'https://icme.stanford.edu/'
    parsed_urls = []

    def parse(self, url: str):
        self.parse_url.append(url)
        for next_url in Website(url):
            self.parse(next_url)
```

```
parsed_urls = ["https://icme.stanford.edu",
                "https://www.stanford.edu/"]
]
```

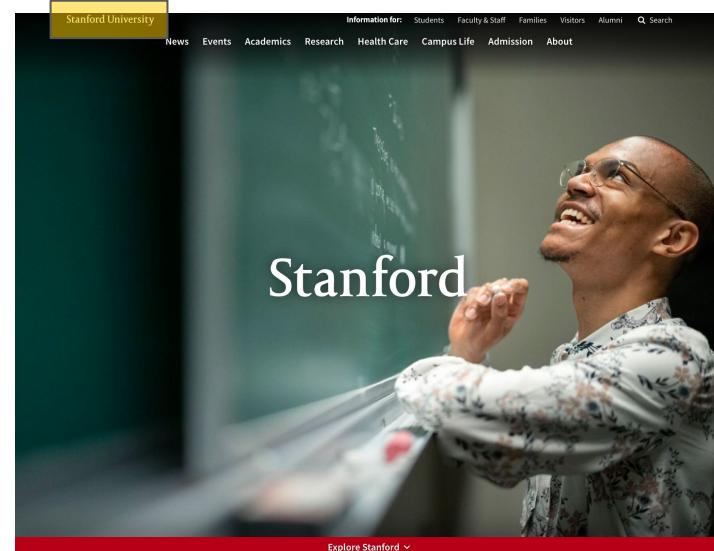


# Search Engine: Web Crawler - Recursion

```
class Spider:
    name = 'icme_spider'
    start_urls = 'https://icme.stanford.edu/'
    parsed_urls = []

    def parse(self, url: str):
        self.parse_url.append(url)
        for next_url in Website(url):
            self.parse(next_url)
```

```
parsed_urls = ["https://icme.stanford.edu",
                "https://www.stanford.edu/"
                ]
```

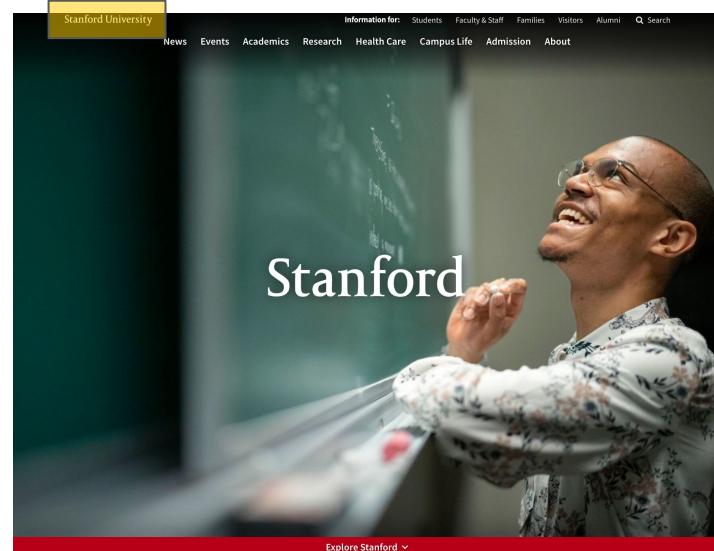


# Search Engine: Web Crawler - Recursion

```
class Spider:  
    name = 'icme_spider'  
    start_urls = 'https://icme.stanford.edu/'  
    parsed_urls = []  
  
    def parse(self, url: str):  
        self.parse_url.append(url)  
        for next_url in Website(url):  
            self.parse(next_url)
```

Oh no, it is “<https://www.stanford.edu/>” again -- we had a bug in the code!

```
parsed_urls = ["https://icme.stanford.edu",  
               "https://www.stanford.edu/"]
```

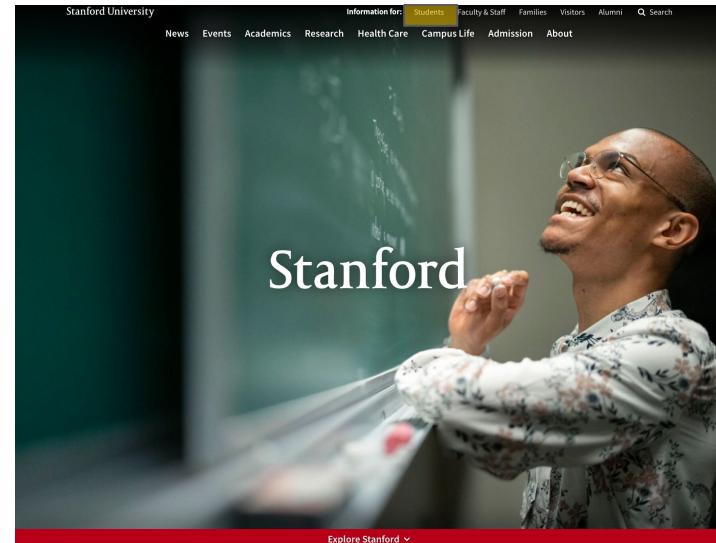


# Search Engine: Web Crawler - Recursion

```
class Spider:
    name = 'icme_spider'
    start_urls = 'https://icme.stanford.edu/'
    parsed_urls = []

    def parse(self, url: str):
        self.parse_url.append(url)
        for next_url in Website(url):
            if next_url not in self.parsed_urls:
                self.parse(next_url)
```

```
parsed_urls = ["https://icme.stanford.edu",
               "https://www.stanford.edu/"]
]
```

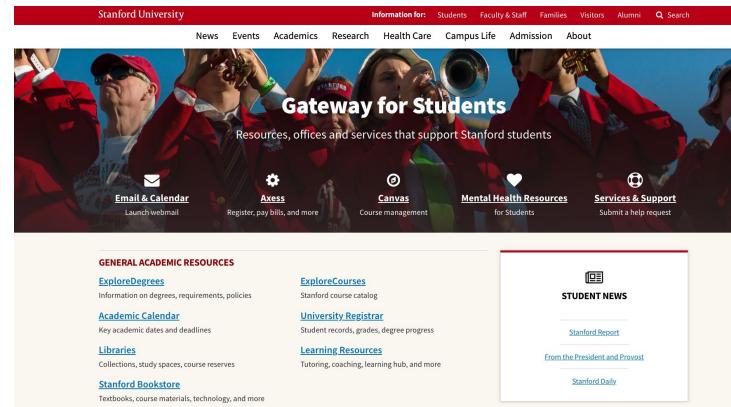


# Search Engine: Web Crawler - Recursion

```
class Spider:
    name = 'icme_spider'
    start_urls = 'https://icme.stanford.edu/'
    parsed_urls = []

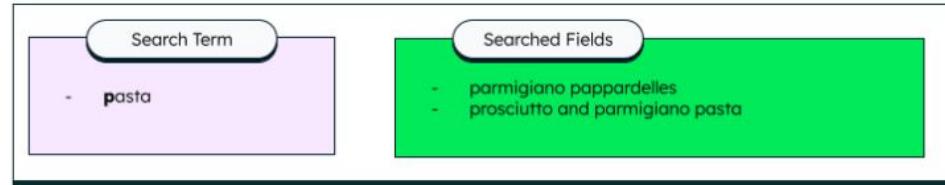
    def parse(self, url: str):
        self.parse_url.append(url)
        for next_url in Website(url):
            if next_url not in self.parsed_urls:
                self.parse(next_url)
```

```
parsed_urls = ["https://icme.stanford.edu",
    "https://www.stanford.edu/"
    "https://www.stanford.edu/student-gateway/"
]
```



- It has a search bar!
  - -> How to incorporate the user intention?
- The items are mal-defined at the first glance.
  - -> How to crawl the internet and store the items?
- User does not simply rate the search results!
  - -> How to assign user-item rating with user data?

# Search Engine: String Search



Average:  $O(n+m)$ ; Worst case:  $O(mn)$

**Rabin-Karp** algorithm, which looks for matching substrings, is fast and easy to implement.

**Knuth-Morris-Pratt** algorithm looks for all instances of a matching character, increasing the speed for multiple matches in a string.

## Search results

This wiki is using a new search engine. ([Learn more](#))

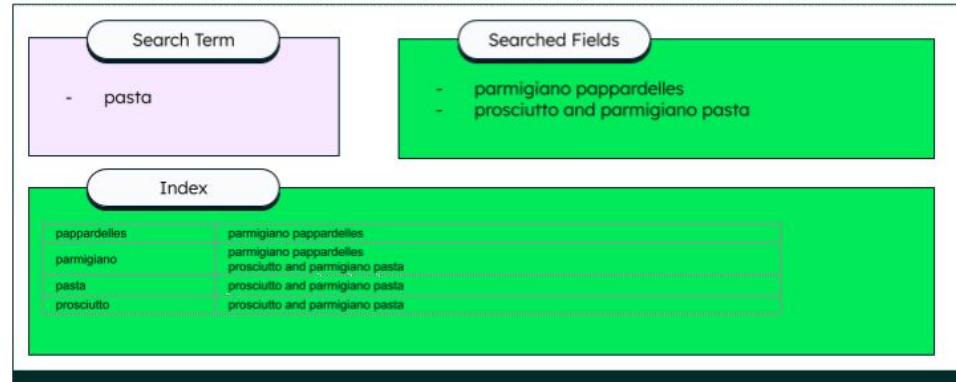
Search

[Content pages](#) [Multimedia](#) [Translations](#) [Everything](#) [Advanced](#)

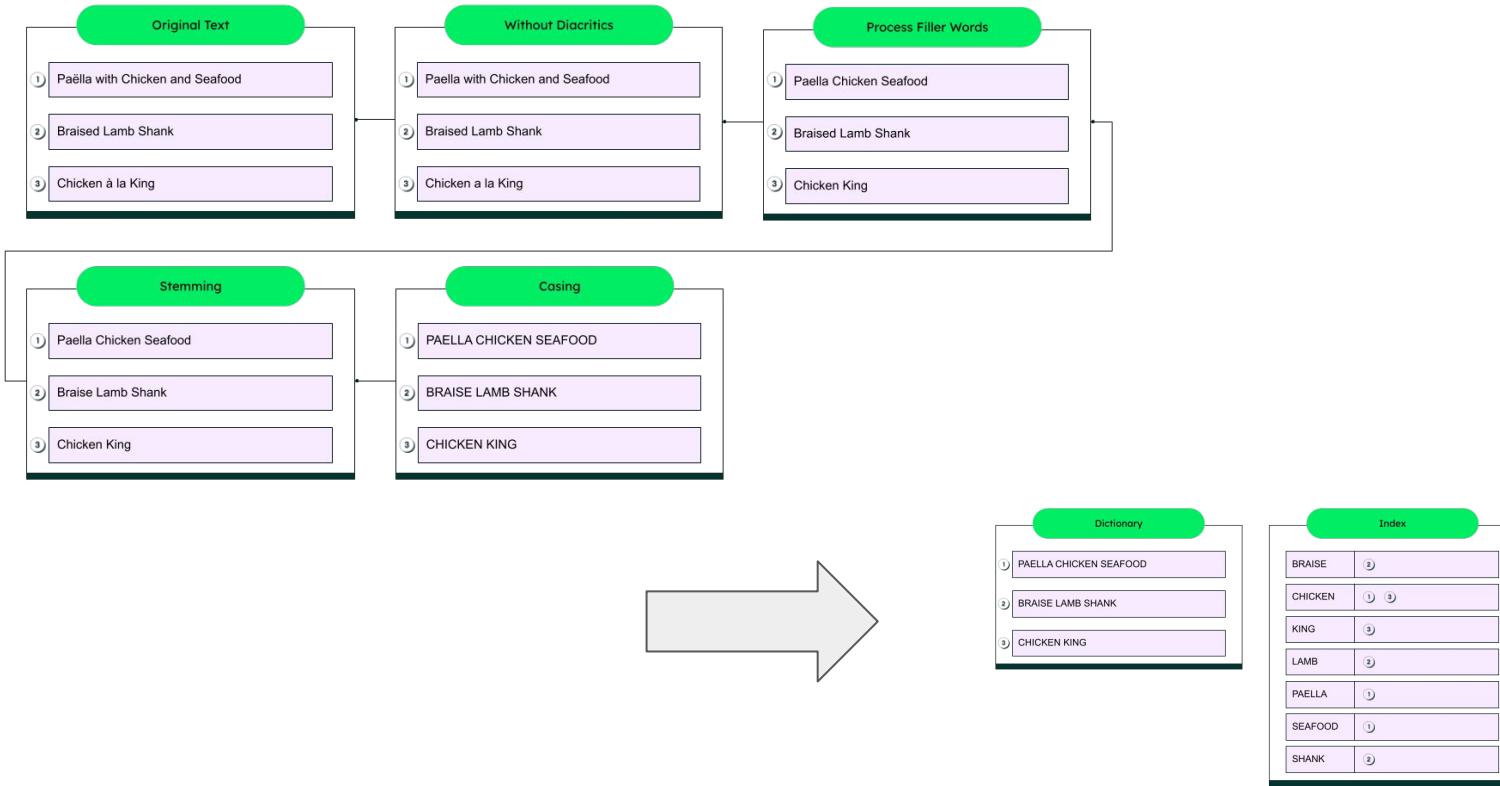
Did you mean: [\*andr  emotions\*](#)

- insertion: *cot* → *coat*
- deletion: *coat* → *cot*
- substitution: *coat* → *cost*

# Search Engine: Full-text Search - Inverted Index



# Search Engine: Full-text Search - Inverted Index



# Search Engine: User-item Rating

- It has a search bar!
  - -> How to incorporate the user intention?
- The items are mal-defined at the first glance.
  - -> How to crawl the internet and store the items?
- User does not simply rate the search results!
  - -> How to assign user-item rating with user data?

## Search Engine: PageRank

- So far, web pages are treated as individual documents.
- But there are hyperlinks between them!

# Search Engine: PageRank

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

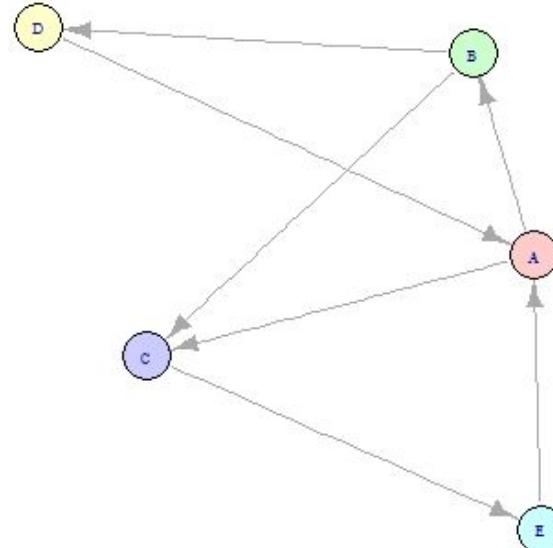
Rank
A 0.2
B 0.2
C 0.2
D 0.2
E 0.2

$p_1, p_2, \dots, p_N$  are the pages under consideration

$M(p_i)$  is the set of pages that link to  $p_i$

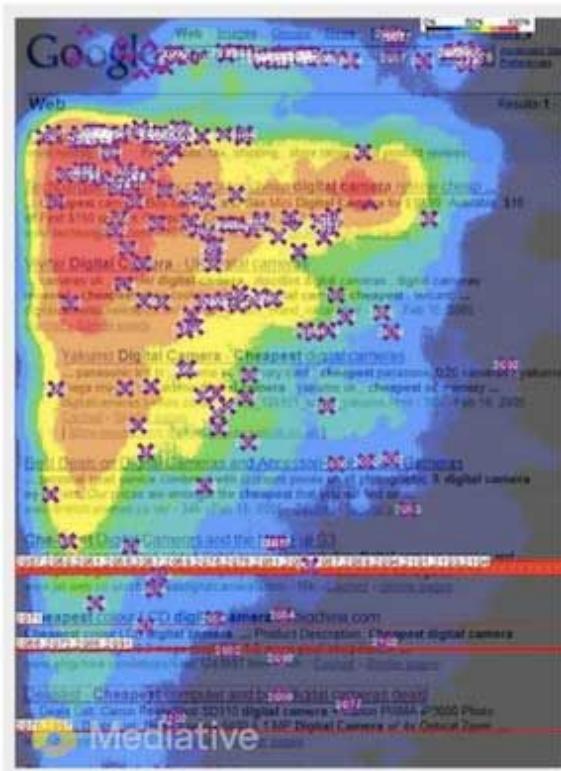
$L(p_j)$  is the number of outbound links on page  $p_j$

Page Rank of the nodes at start



- We have millions of user clicking on some Search Engine Results every day.
- Can we assign click or not as ratings?
  - Yes and no

# Search Engine: User-item Rating - Clicks



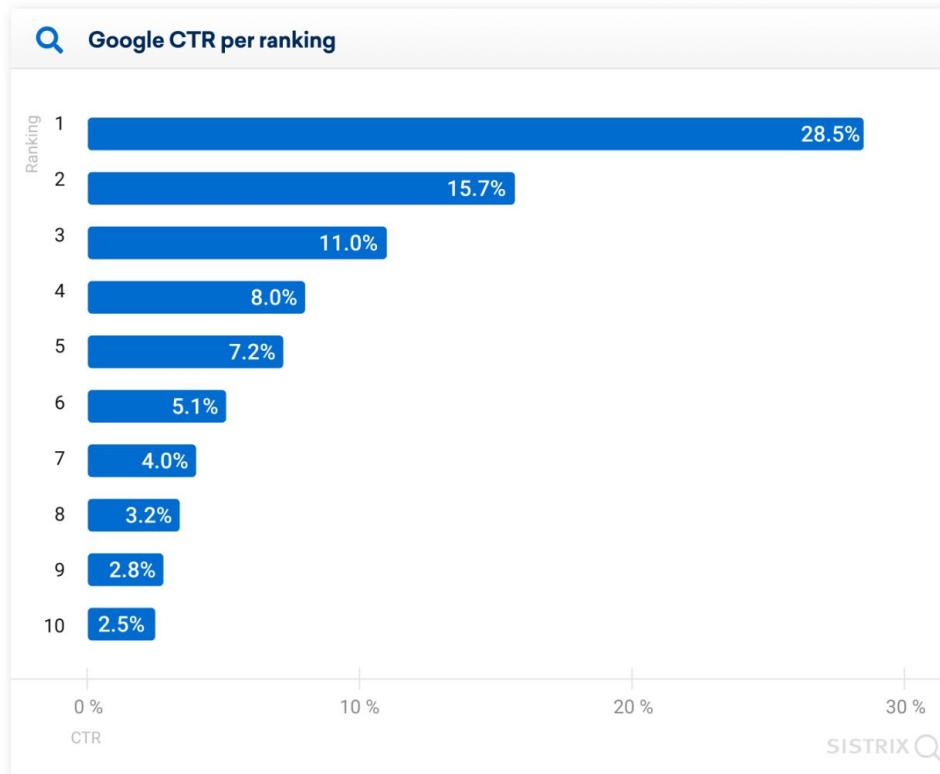
2005



2014

Source: *The Evolution of Google Search Results Pages*, Mediative, 2014

# Search Engine: User-item Rating - Clicks



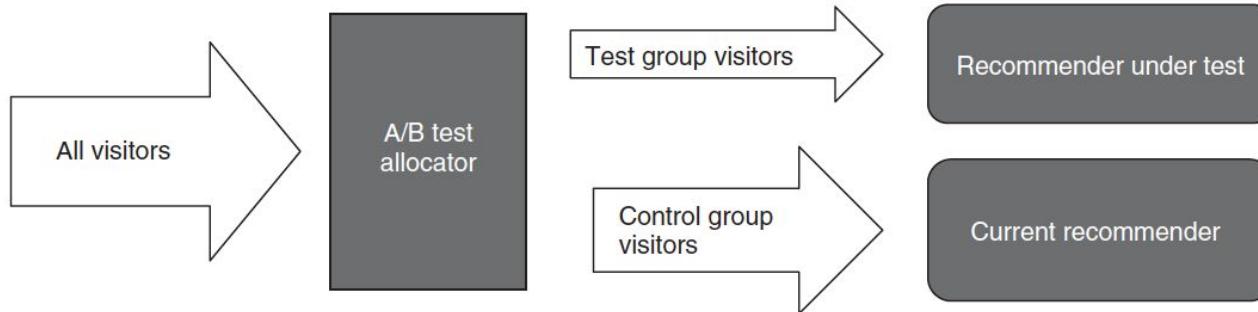
## Search Engine: Evaluation - Clicks

$$\text{CTR} = \frac{\text{CLICKS}}{\text{IMPRESSIONS}} \times 100$$

**CLICKS**  
Number of people who clicked the ad

**IMPRESSIONS**  
Number of people who saw the ad

# Search Engine: Recap of Online Evaluation



**Figure 9.16** In an A/B test, visitors are split into two groups: the test group that sees the new feature and a control group that continues as usual.

- Q: For Search Engine, how to measure the user satisfaction and success?
  - Any potential bias?
- Q: How about the recommendation system you chose yesterday?



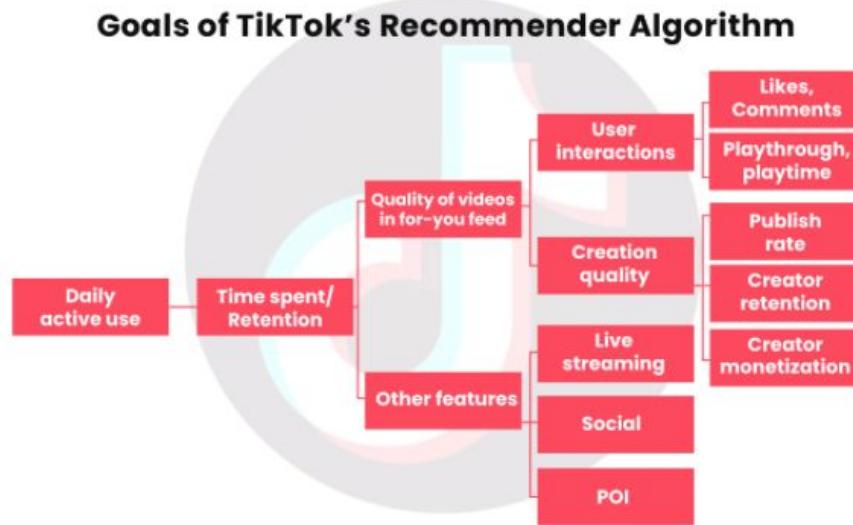
# Search Engine: Time-to-long-click (TTLC)

- Long-click: When a user performs a search, clicks through on a result and remains on that site for a long time.
  - Anti-pattern: Pogo-sticking
- Domain specific
- Knowledge panels and direct answers



MOM

# Tiktok: Multi-target



# Advanced Topics of Recommendation System

IF BRUTE FORCE DOESN'T  
SOLVE YOUR PROBLEMS

THEN YOU AREN'T  
USING ENOUGH

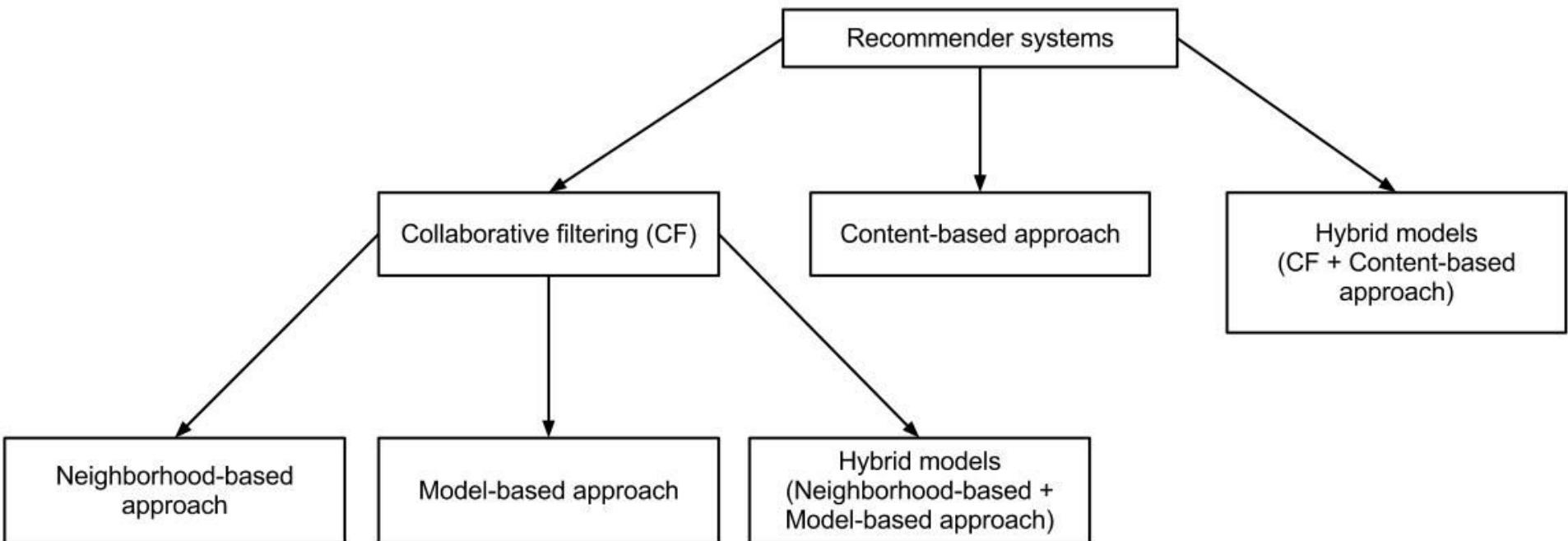
- Deep learning
- Scale up and speed up
- Cold start
- Responsive
- Social impact

- Deep learning

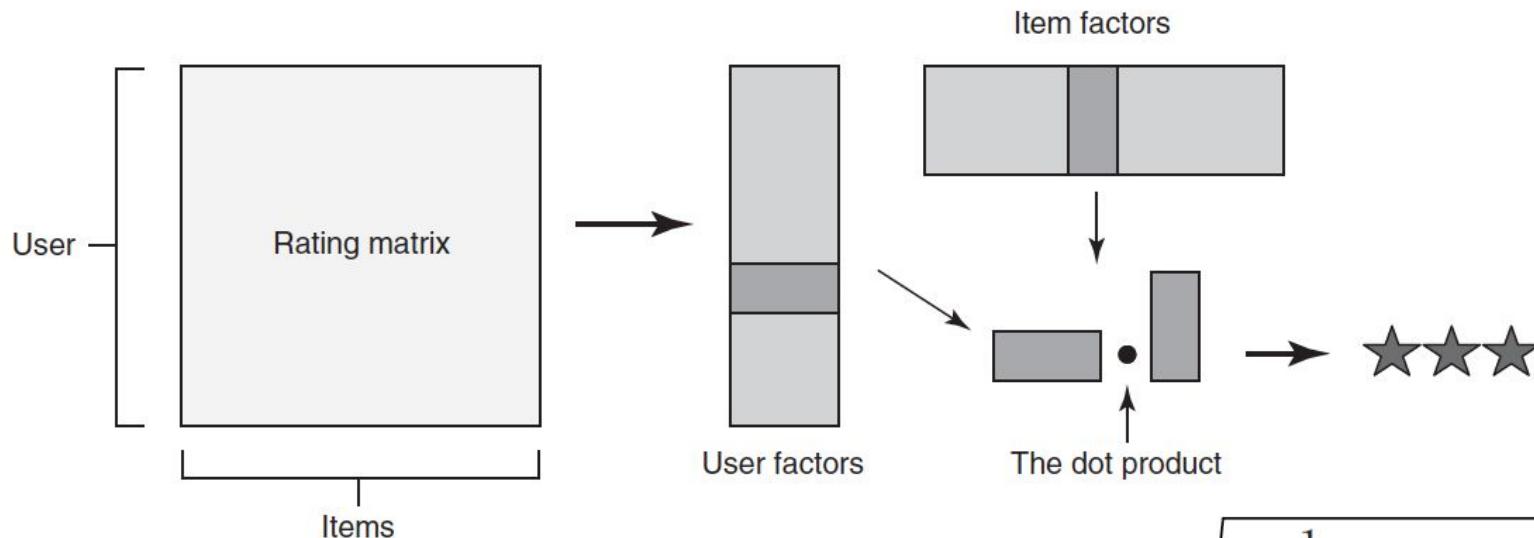
**Break!**

- Scale up and speed up
- LLM
- Social impact

# Recommendation System w/ Deep Learning: Recap



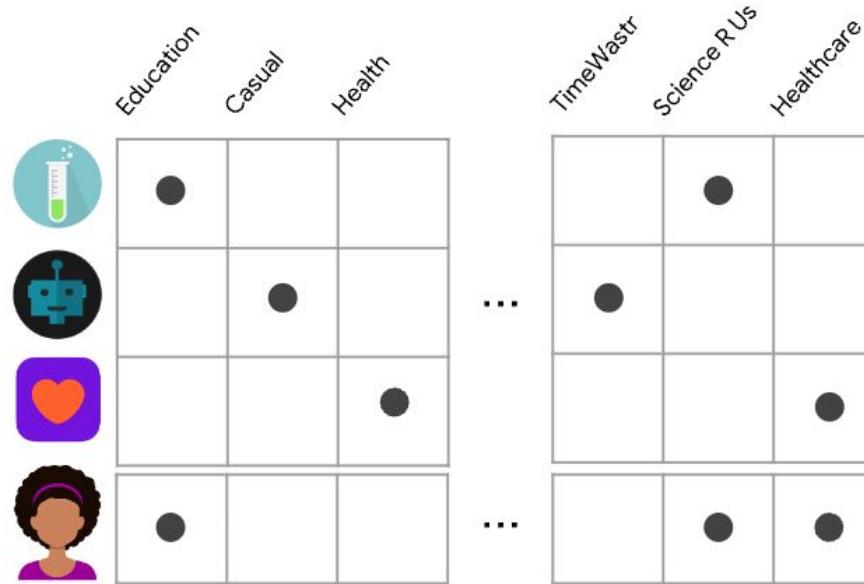
# Recommendation System w/ Deep Learning: Recap



$$RMSE = \sqrt{\frac{1}{|known|} \sum_{(u,i) \in known} (r_{ui} - u_u v_i)^2}$$

- Rating/score can be modeled as a product of user vector and item vector.

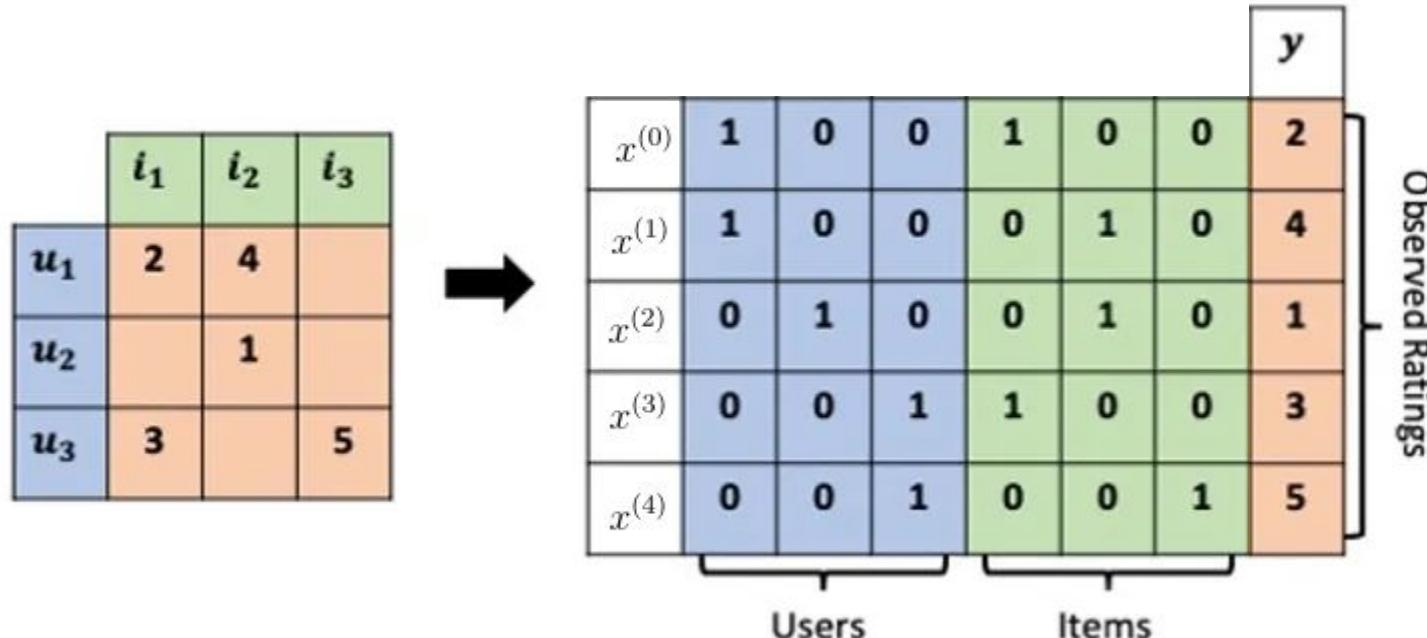
# Recommendation System w/ Deep Learning: Recap



- There are well-defined features (of both users and items) that can be used for the prediction.

- Factorization Machines (FM): Let's combine the best of the both worlds!

# Reformulate Collaborative Filtering



# Reformulate Collaborative Filtering

	$i_1$	$i_2$	$i_3$
$u_1$	2	4	
$u_2$		1	
$u_3$	3		5



$x^{(0)}$	1	0	0	1	0	0	2
$x^{(1)}$	1	0	0	0	1	0	4
$x^{(2)}$	0	1	0	0	1	0	1
$x^{(3)}$	0	0	1	1	0	0	3
$x^{(4)}$	0	0	1	0	0	1	5

Users                    Items

Observed Ratings

$$\hat{r}_{1,1} = w_0 + u_1^T v_1$$

$$\hat{r}_{1,1} = w_0 + \left( \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} (u_1, u_2, u_3) \right)^T \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} (v_1, v_2, v_3)$$

# Reformulate Collaborative Filtering

	$i_1$	$i_2$	$i_3$
$u_1$	2	4	
$u_2$		1	
$u_3$	3		5



Users                    Items

$x^{(0)}$	1	0	0	1	0	0	2
$x^{(1)}$	1	0	0	0	1	0	4
$x^{(2)}$	0	1	0	0	1	0	1
$x^{(3)}$	0	0	1	1	0	0	3
$x^{(4)}$	0	0	1	0	0	1	5

Observed Ratings

$$\hat{r}_{1,1} = w_0 + u_1^T v_1$$

$$\hat{r}_{1,1} = w_0 + \sum_{i=1}^n \sum_{j=i+1}^n x_i^{(0)} x_j^{(0)} u_i^T v_j$$

# Reformulate Content Filtering

	$i_1$	$i_2$	$i_3$
$u_1$	2	4	
$u_2$		1	
$u_3$	3		5



E.g. News Popularity

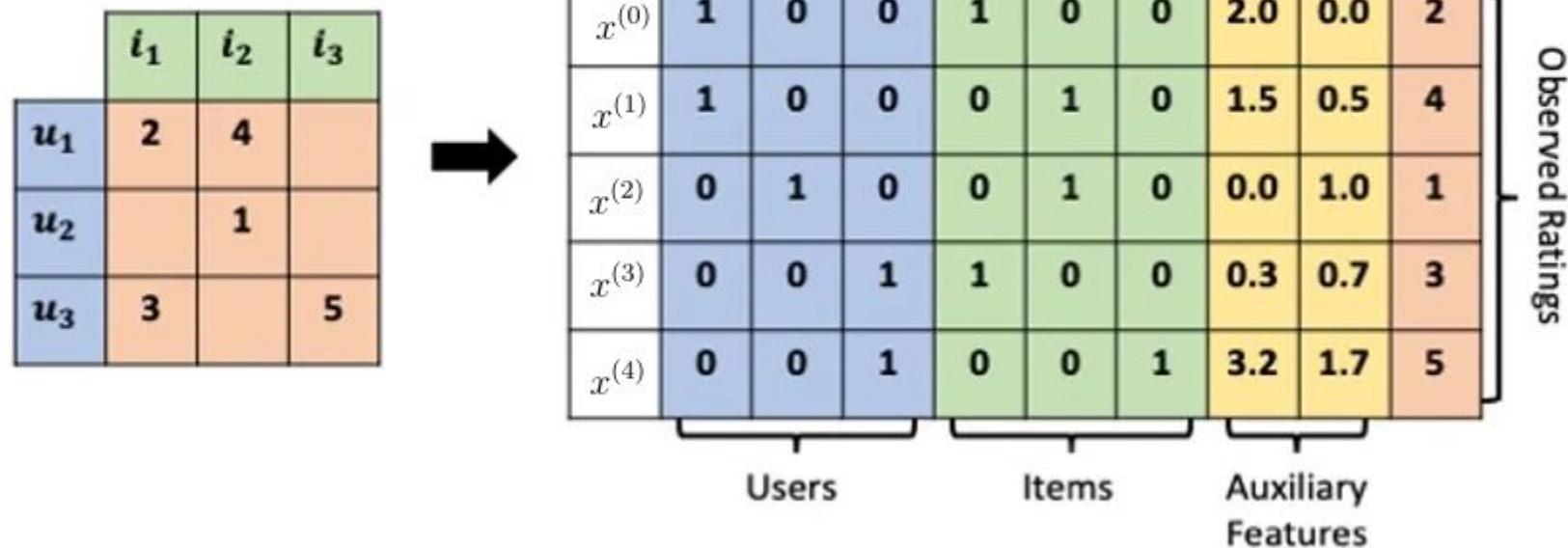
	$i_1$	$i_2$	$i_3$	$a_1$	$a_2$	$y$
$x^{(0)}$	1	0	0	2.0	0.0	2
$x^{(1)}$	0	1	0	1.5	0.5	4
$x^{(2)}$	0	1	0	1.5	0.5	1
$x^{(3)}$	1	0	0	2.0	0.0	3
$x^{(4)}$	0	0	1	3.2	1.7	5

Observed Ratings

Items      Auxiliary Features

$$\hat{r}_{1,1} = w_0 + w_1 a_1 + w_2 a_2$$

# Back to Factorization Machines (FM)

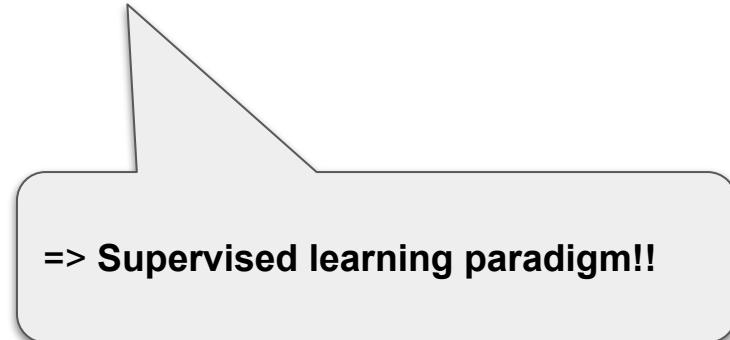


$$\hat{r} = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n x_i x_j v_i^T v_j$$

## Back to Factorization Machines (FM)

$$\hat{r} = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n x_i x_j v_i^T v_j$$

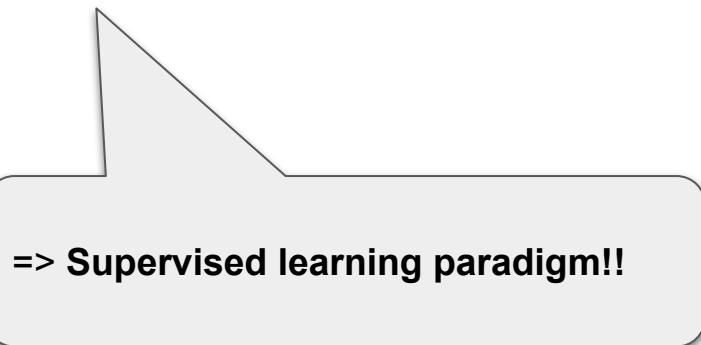
- $w_0$ : Global bias
- $w_i$ : Weights of features
- $v_i$ : Feature factor  $i$



# Back to Factorization Machines (FM)

$$\hat{r} = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n x_i x_j v_i^T v_j$$

- $w_0$ : Global bias
- $w_i$ : Weights of features
- $v_i$ : Feature factor i



Rating Matrix



Collaborative Filtering



FM

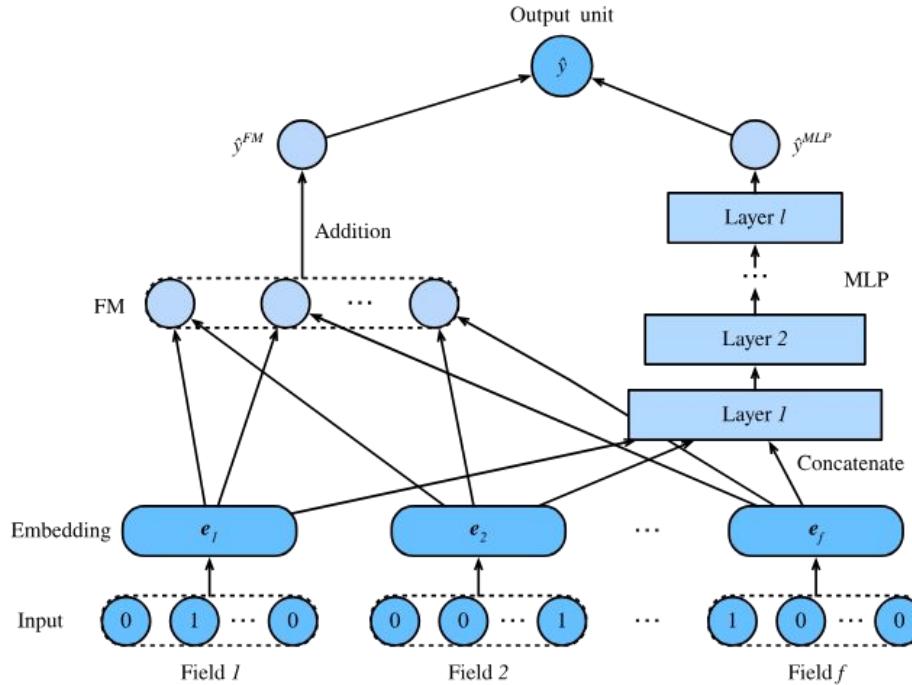


Welcome to the  
Deep-world

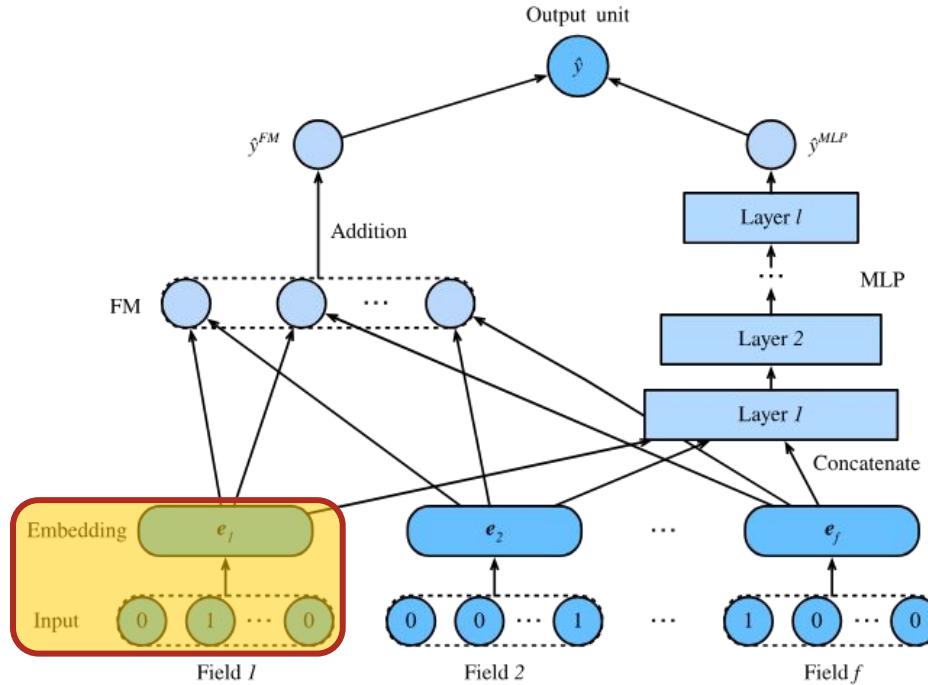


# Factorization Machines (FM): Pros & Cons

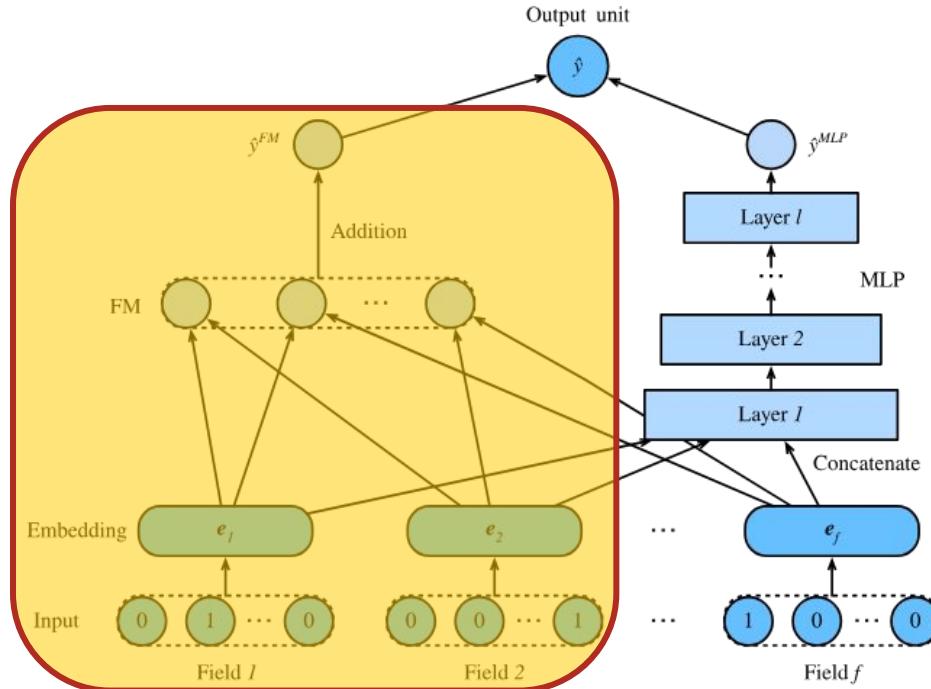
- Capture user-item feature interaction
- Efficient for sparse data
- Non-linear patterns

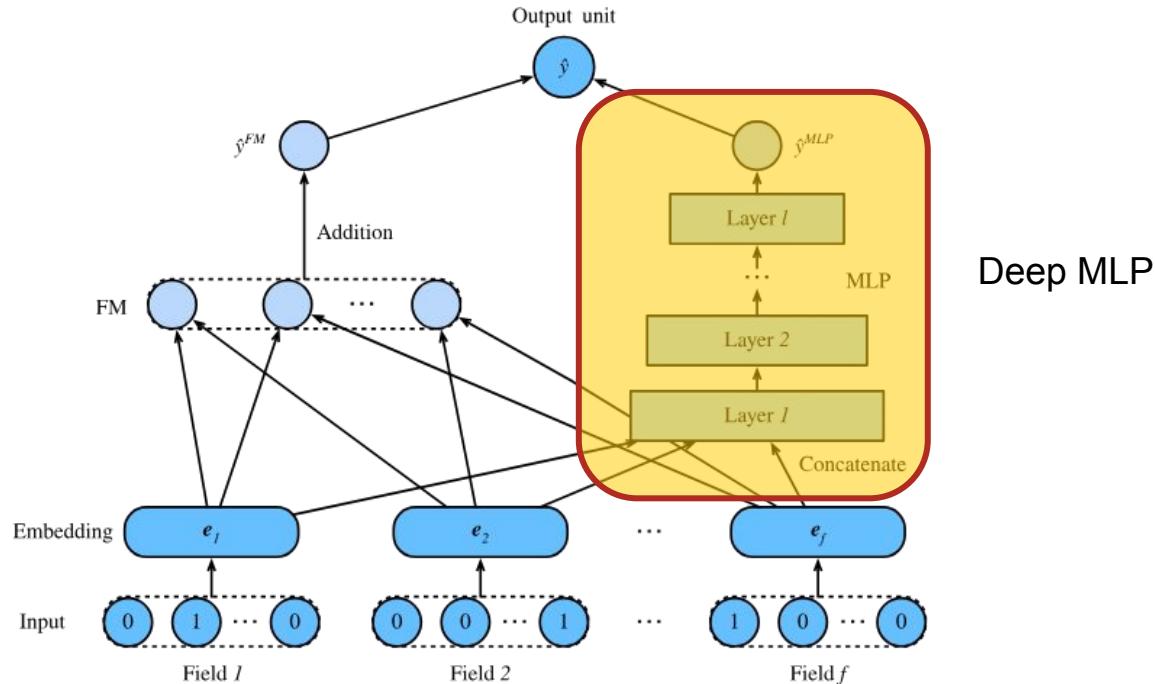


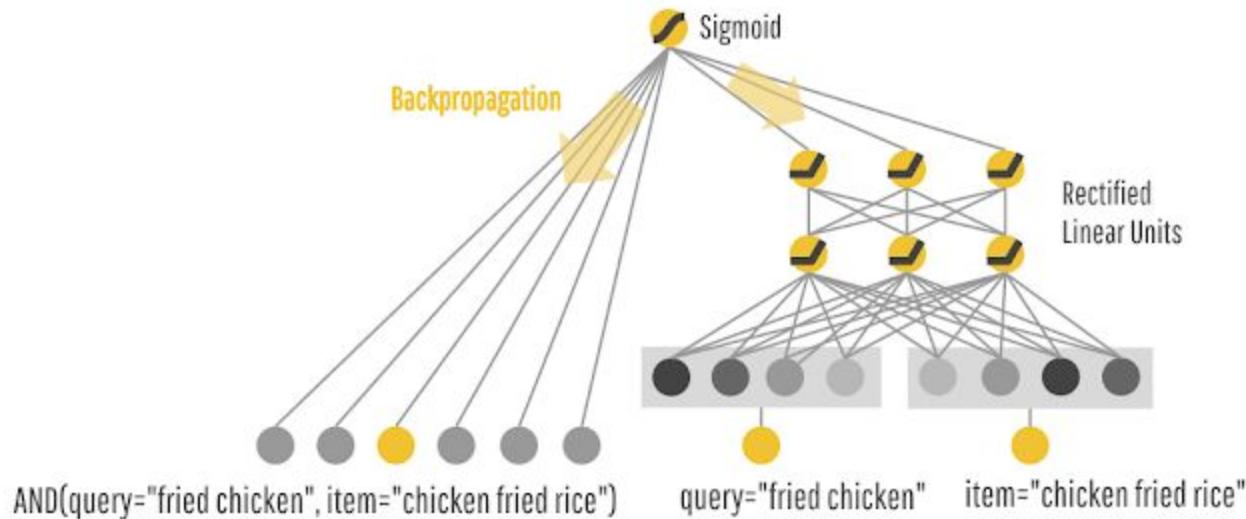
Factors =  
Embeddings



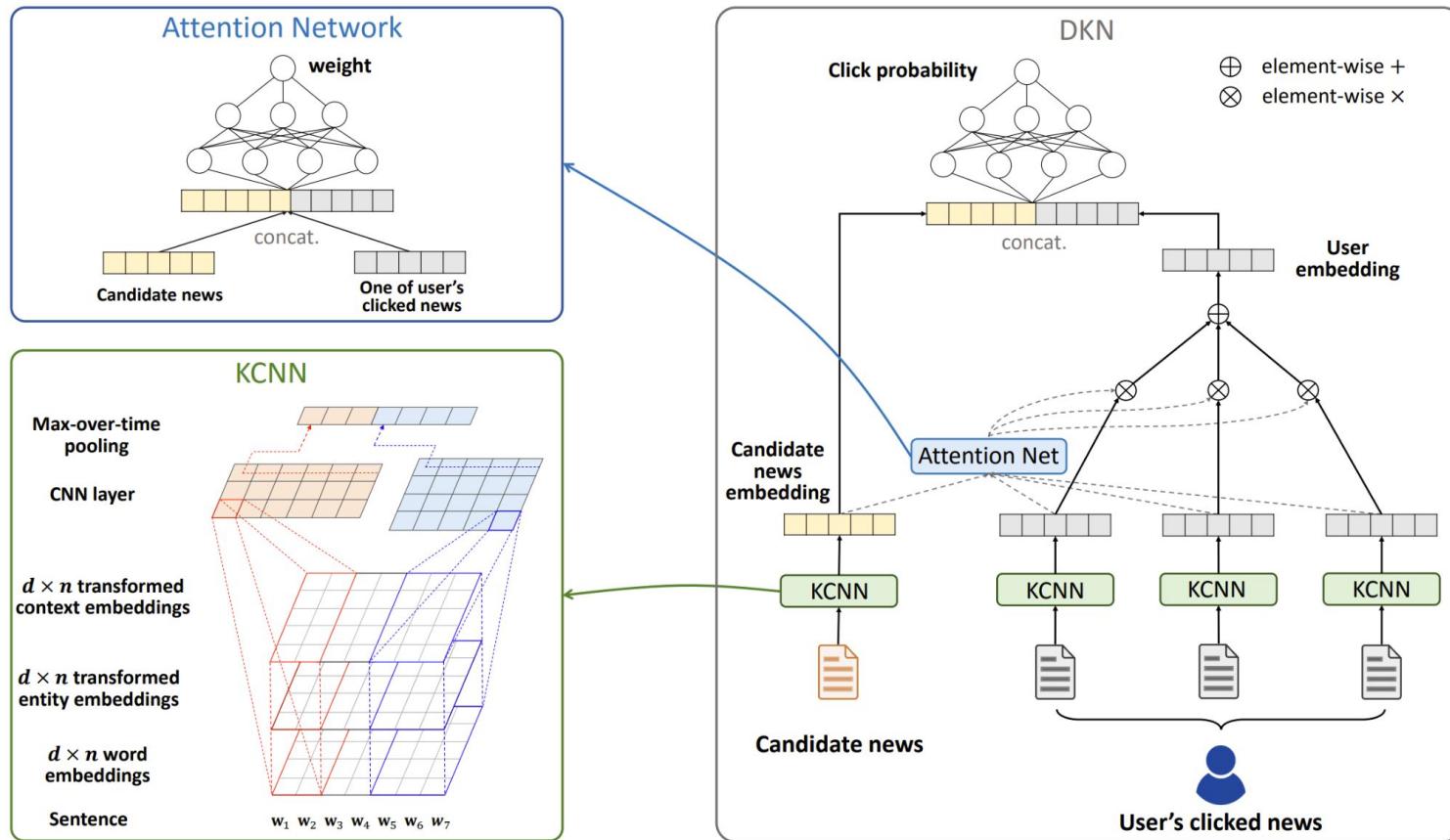
## Factorization Machines







# Deep Knowledge-Aware Network (2018)

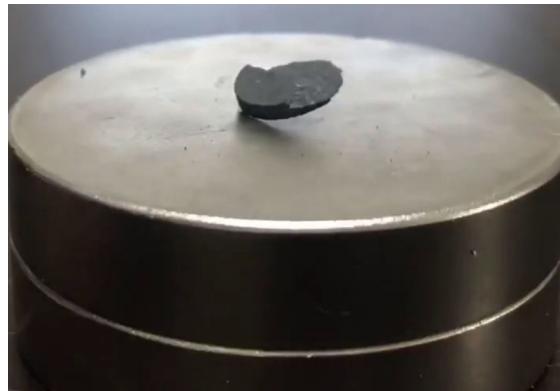


- Sparse Features: Tag, device, etc.
- Dense Features: Age, income, # of videos watched, etc.
- **Sparse Features** are the first-class citizens.

- **Sparse Features** are the first-class citizens.
  - Dense feature requires more parameterization.
    - Age -> like this YouTube Video
      - Assumption: Age = 9 -> Age = 10 has a same effect of Age = 43 -> Age = 44

- **Sparse Features** are the first-class citizens.
  - Dense feature requires more parameterization.
  - Sparse features are easy to generate cross-features.
    - Hashing tricks: Only keep K more frequent tuples of the combinations.

#lk99 #superconductor



#lk99 #drinking-solution



LK99

Elkay Deluxe 3-1/2" Drain Type 304 Stainless Steel Body St  
Rubber Seal and Tailpiece

\$169.00 (USD)

Actual selling price may vary

[Where to Buy](#) [Contact Us](#)

★★★★★ 4.7 | 20 Reviews

14 out of 16 (83%) reviewers recommend this product

14 questions and 14 answers for this product

[WRITE A REVIEW](#)

Share:    

[Downloads](#)

 [Specification Sheet \(pdf\)](#)

 [Care Cleaning \(pdf\)](#)

- **Sparse Features** are the first-class citizens.
  - Dense feature requires more parameterization.
  - Sparse features are easy to generate cross-features.
  - \* Easy for online training/serving

# Lab Time!



# Advanced Topics (Part II)



Hours of watching randomly recommended vids

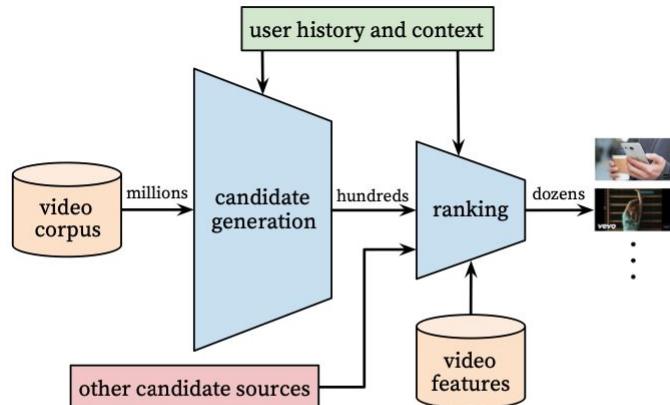
- Deep learning
- Scale up and speed up
- Cold start
- Responsive
- Social impact

## Scale up & Speed up: Speed Matters

*All other things being equal ... our experiments demonstrate that slowing down the search results page by **100 to 400 milliseconds** has a measurable impact on the number of searches per user of **-0.2%** to **-0.6%**. That's 0.2% to 0.6% fewer searches for changes under half a second!*

-- Google Research Blog

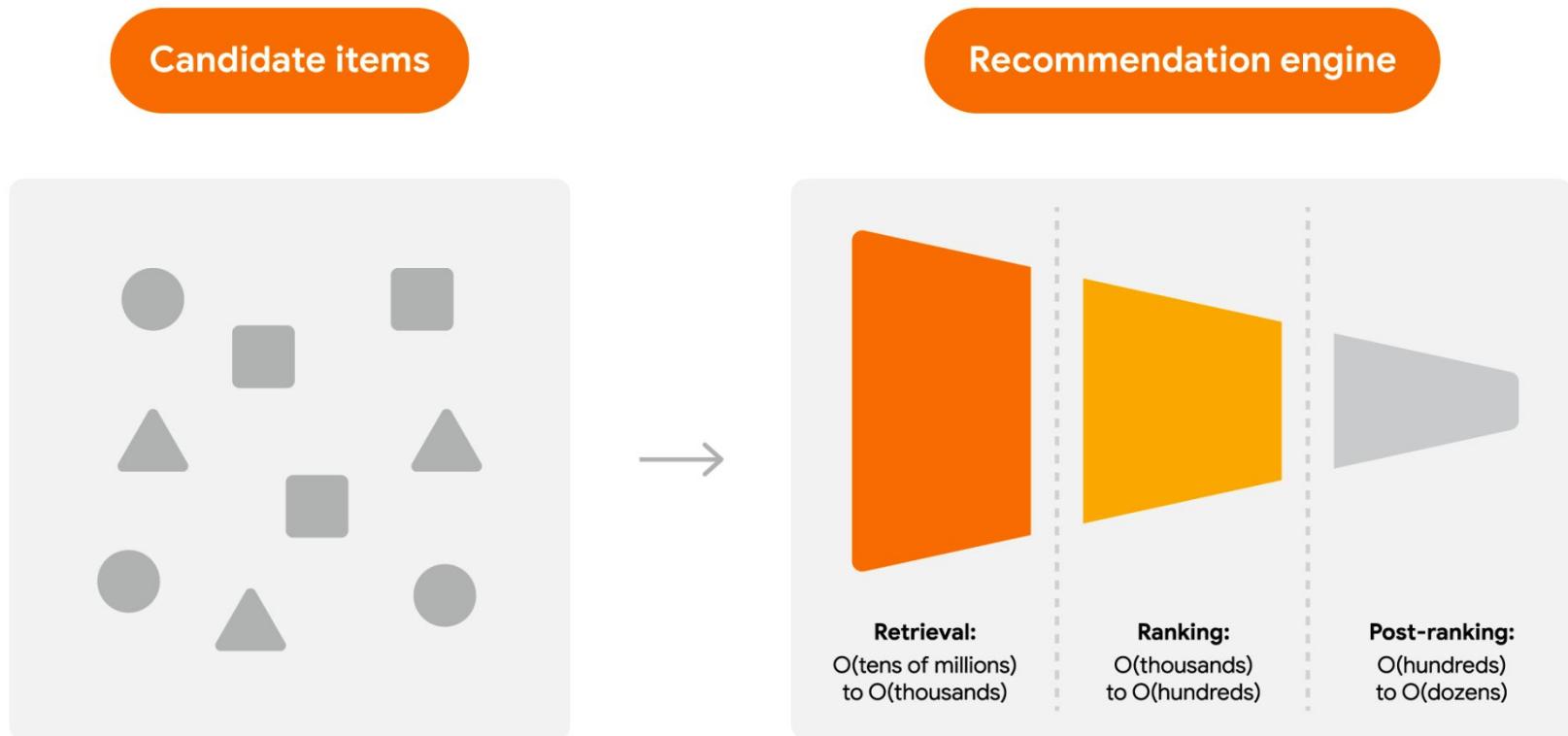
# Speed-up: Recommendation Funnel



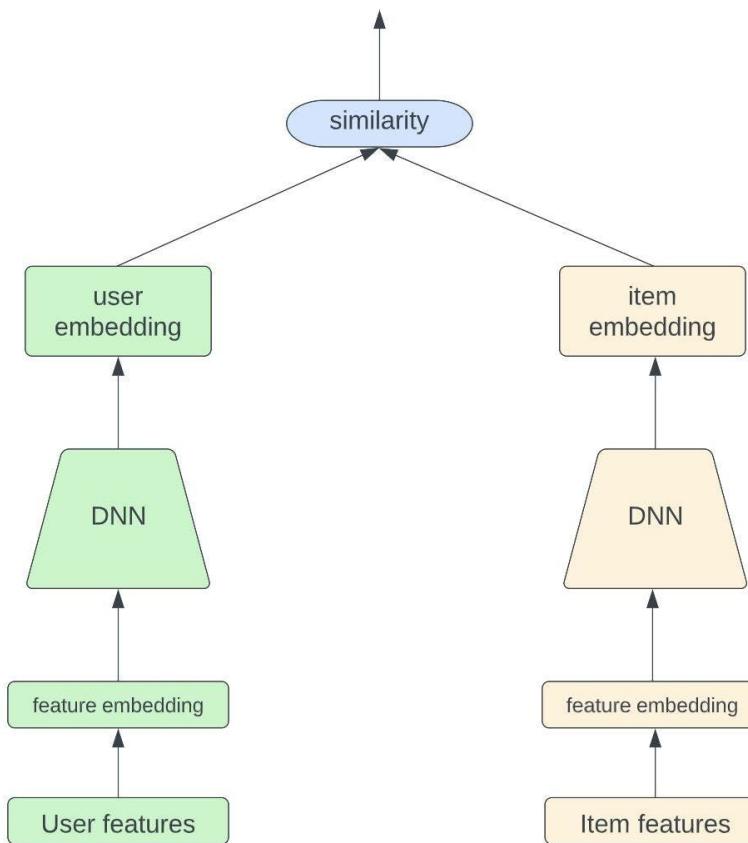
- Retrieval / Candidate-gen
- Ranking / Sort
- Re-rank

Figure 2: Recommendation system architecture demonstrating the “funnel” where candidate videos are retrieved and ranked before presenting only a few to the user.

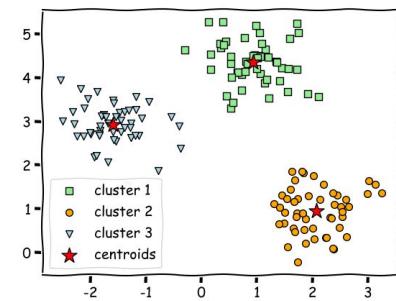
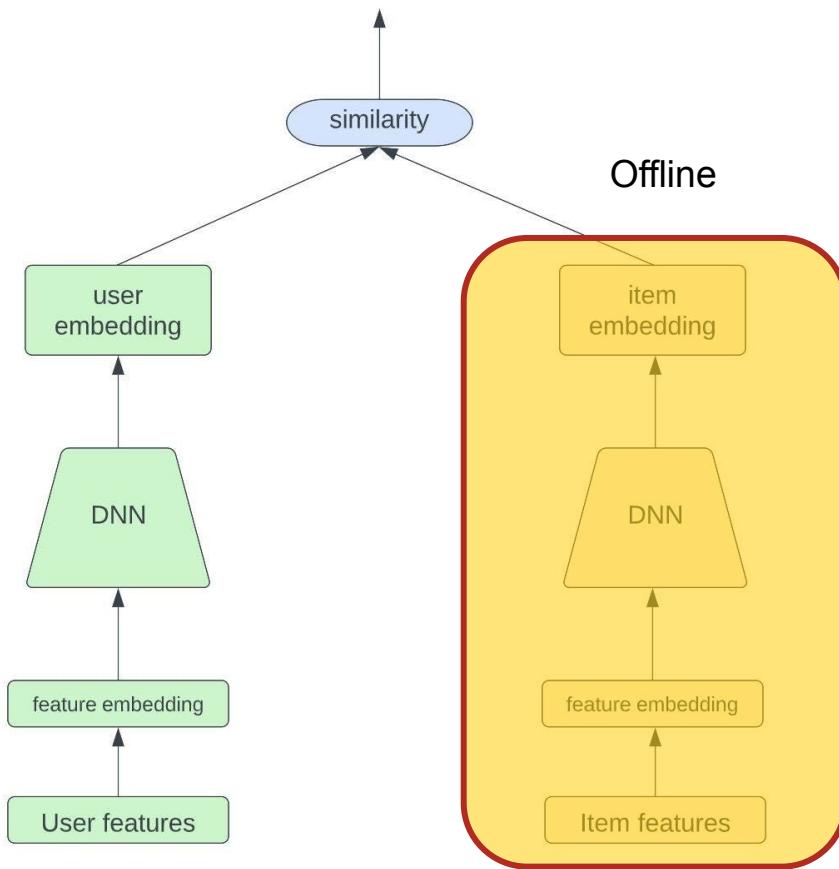
# Speed-up: Recommendation Funnel



# Speed-up: Two-tower Retrieval

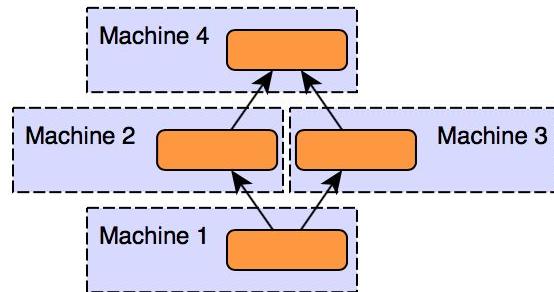


# Speed-up: Two-tower Retrieval

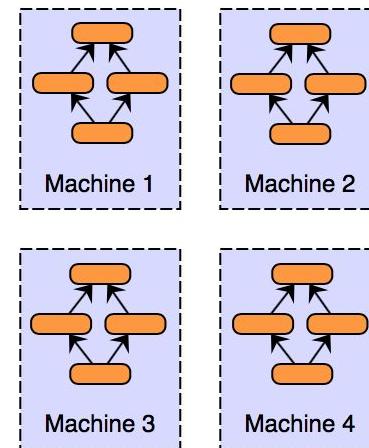


# Scale-up: Parameter Server (PS)

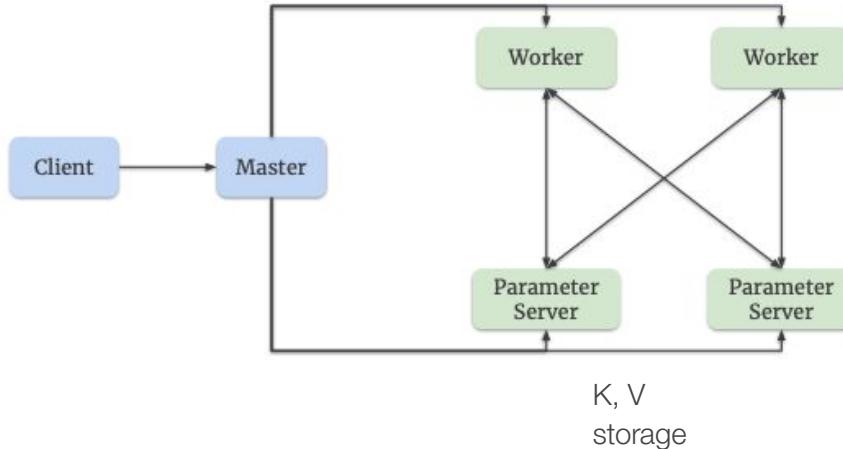
Model Parallelism



Data Parallelism



# Scale-up: Parameter Server (PS)




---

## Algorithm 1 Distributed Subgradient Descent

### Task Scheduler:

```

1: issue LoadData() to all workers
2: for iteration  $t = 0, \dots, T$  do
3:   issue WORKERITERATE( $t$ ) to all workers.
4: end for

```

### Worker $r = 1, \dots, m$ :

```

1: function LOADDATA()
2:   load a part of training data  $\{y_{i_k}, x_{i_k}\}_{k=1}^{n_r}$ 
3:   pull the working set  $w_r^{(0)}$  from servers
4: end function
5: function WORKERITERATE( $t$ )
6:   gradient  $g_r^{(t)} \leftarrow \sum_{k=1}^{n_r} \partial \ell(x_{i_k}, y_{i_k}, w_r^{(t)})$ 
7:   push  $g_r^{(t)}$  to servers
8:   pull  $w_r^{(t+1)}$  from servers
9: end function

```

### Servers:

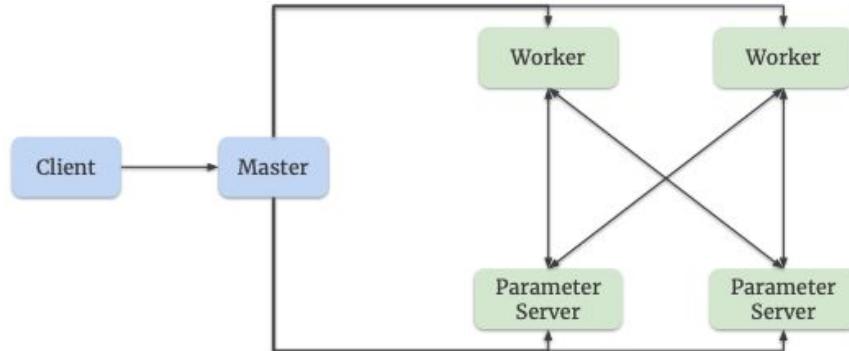
```

1: function SERVERITERATE( $t$ )
2:   aggregate  $g^{(t)} \leftarrow \sum_{r=1}^m g_r^{(t)}$ 
3:    $w^{(t+1)} \leftarrow w^{(t)} - \eta (g^{(t)} + \partial \Omega(w^{(t)}))$ 
4: end function

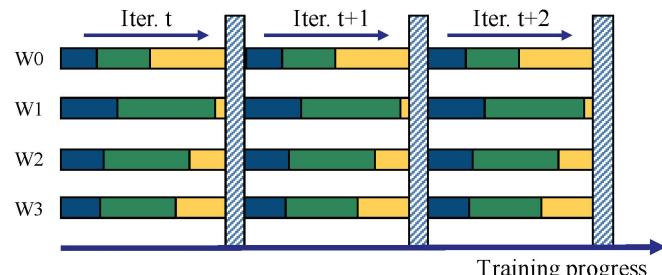
```

---

# Scale-up: Parameter Server (PS)

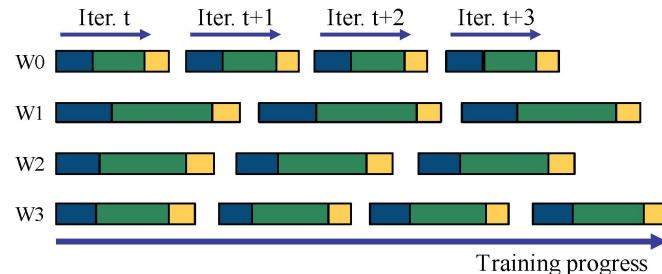


Forward Backward Wait Sync. barrier



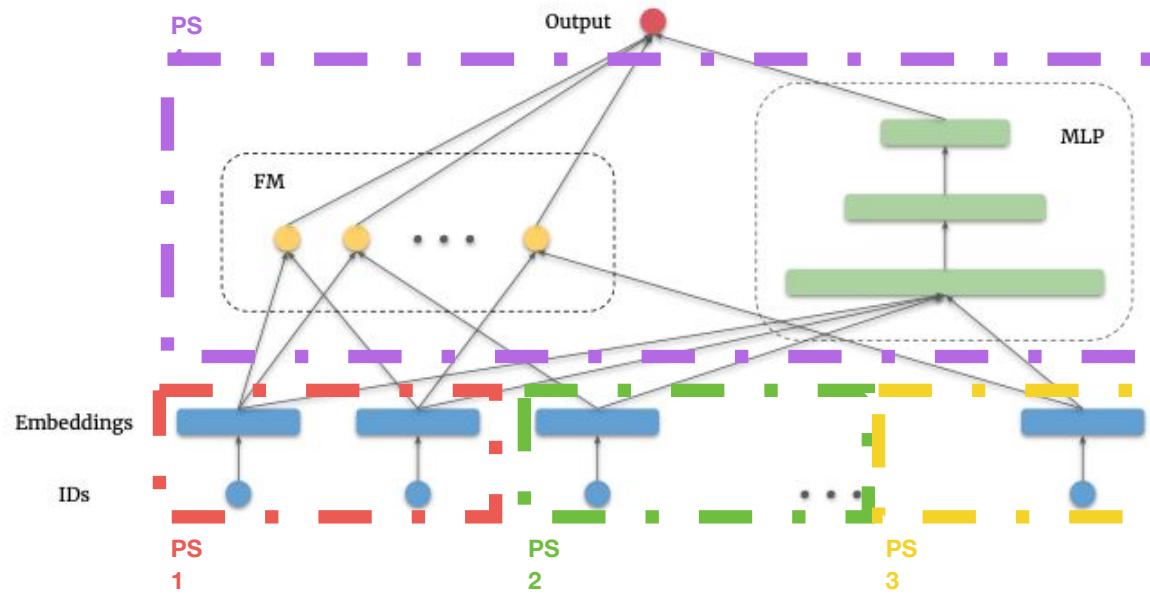
(a)

Forward Backward Async. Comm

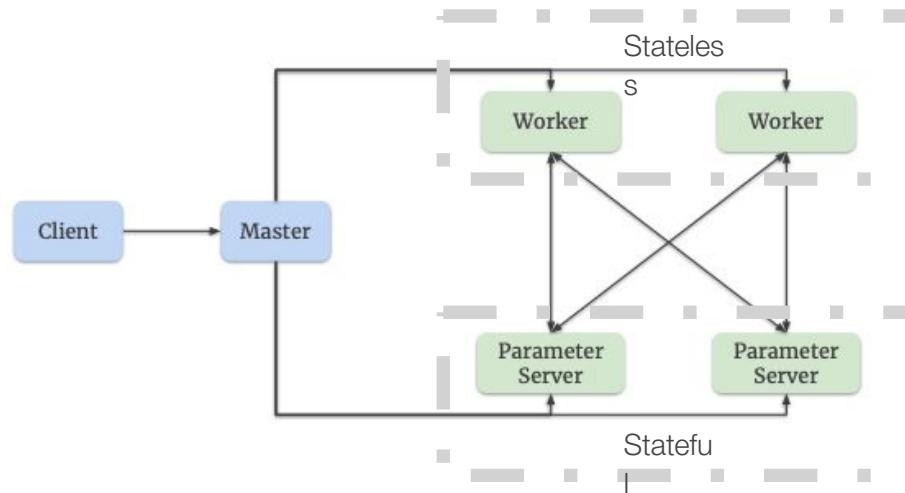


(b)

# Scale-up: Parameter Server (PS)

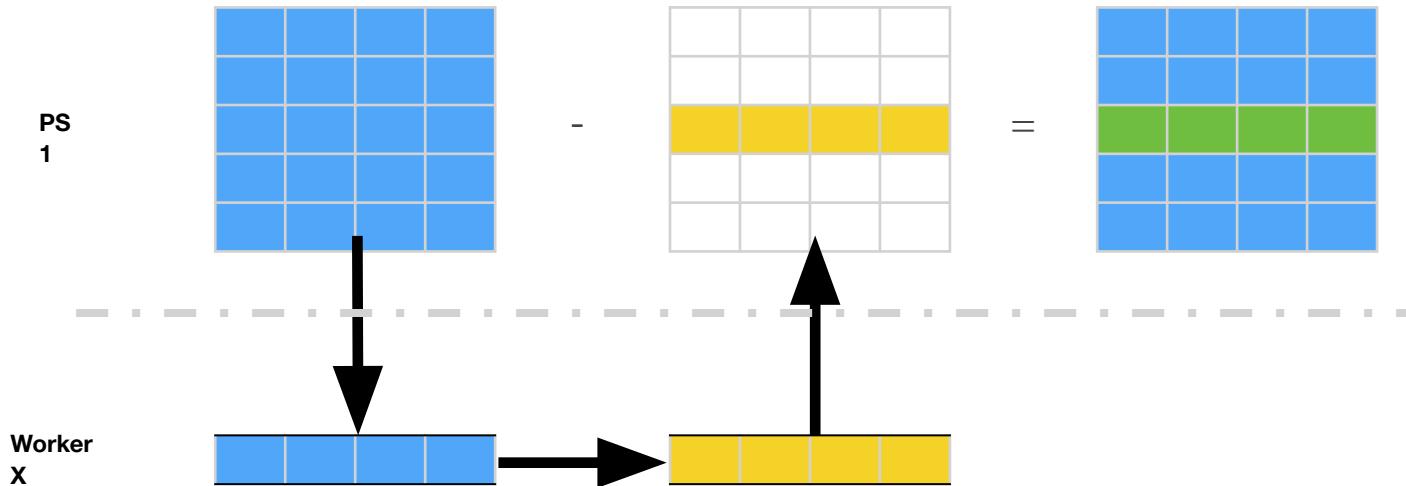


# Scale-up: Parameter Server (PS)



Li, Mu, et al. "Communication efficient distributed machine learning with the parameter server." Advances in Neural Information Processing Systems 27 (2014).

# Scale-up: Parameter Server (PS)

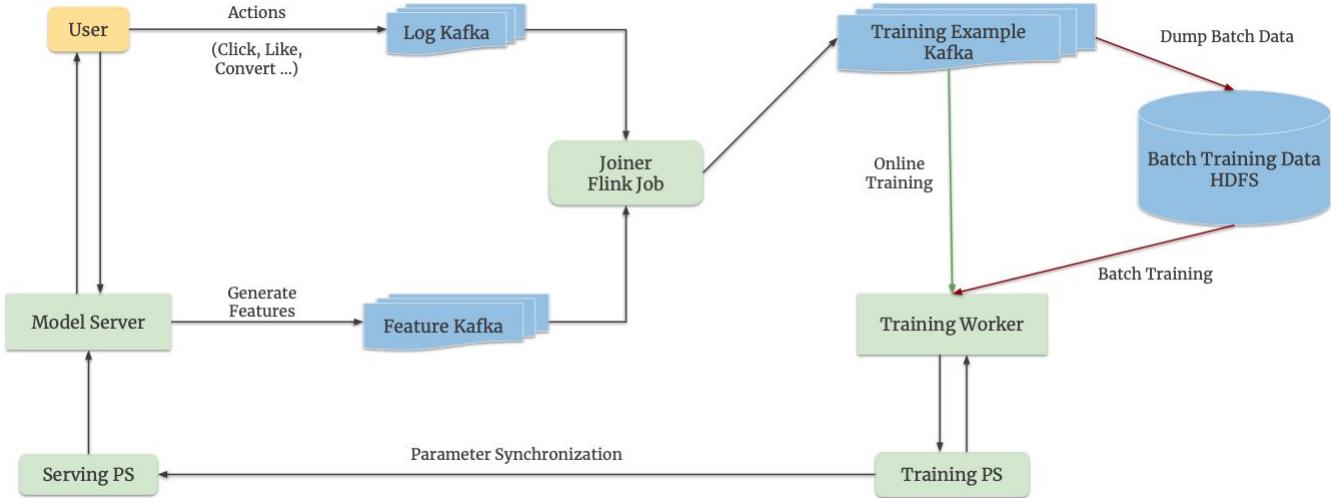




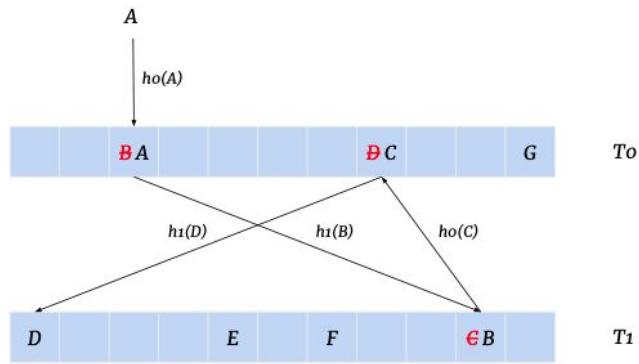
## Augmenting recommendation systems with LLMs (Large Language Model)

when you need advice but aren't sure who to trust

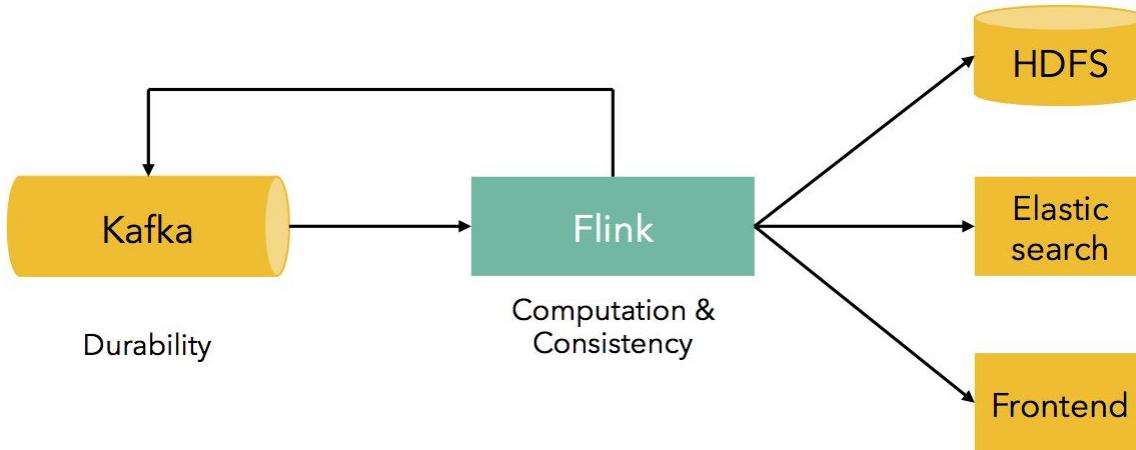




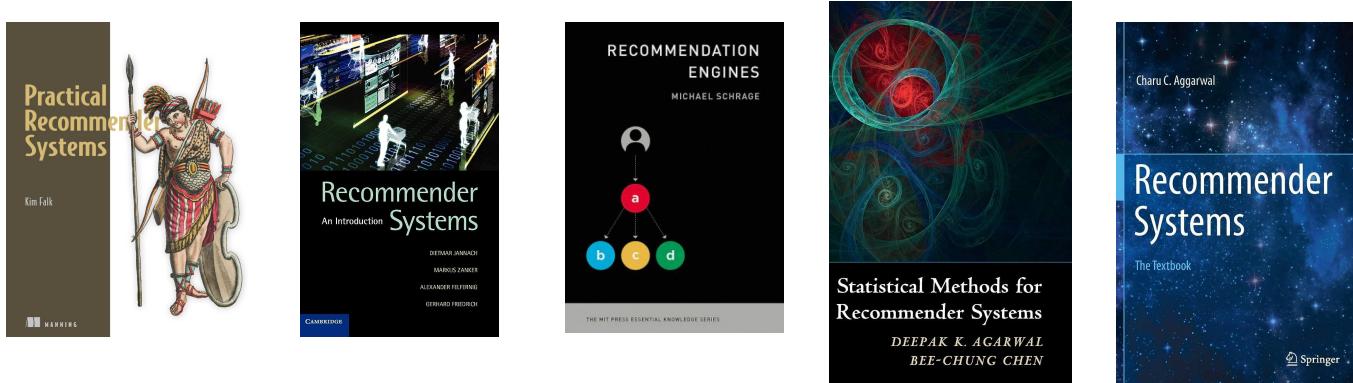
Liu, Zhuoran, et al. "Monolith: Real Time Recommendation System With Collisionless Embedding Table." arXiv preprint arXiv:2209.07663 (2022).



**Figure 3: Cuckoo HashMap.**



# Reference



	PRS	IRS	RE	SERS	RST
Related Chap.	Chap. 7-11.	Chap. 2, 7	Chap. 1-5.	Chap. 1, 2, 4.	Chap 1-4.
Hao's Rating	5	3	4	2	4

**RECOMMENDER SYSTEMS**



**RECOMMENDER SYSTEMS  
EVERYWHERE**

**SHUT UP AND TAKE MY MONEY!**

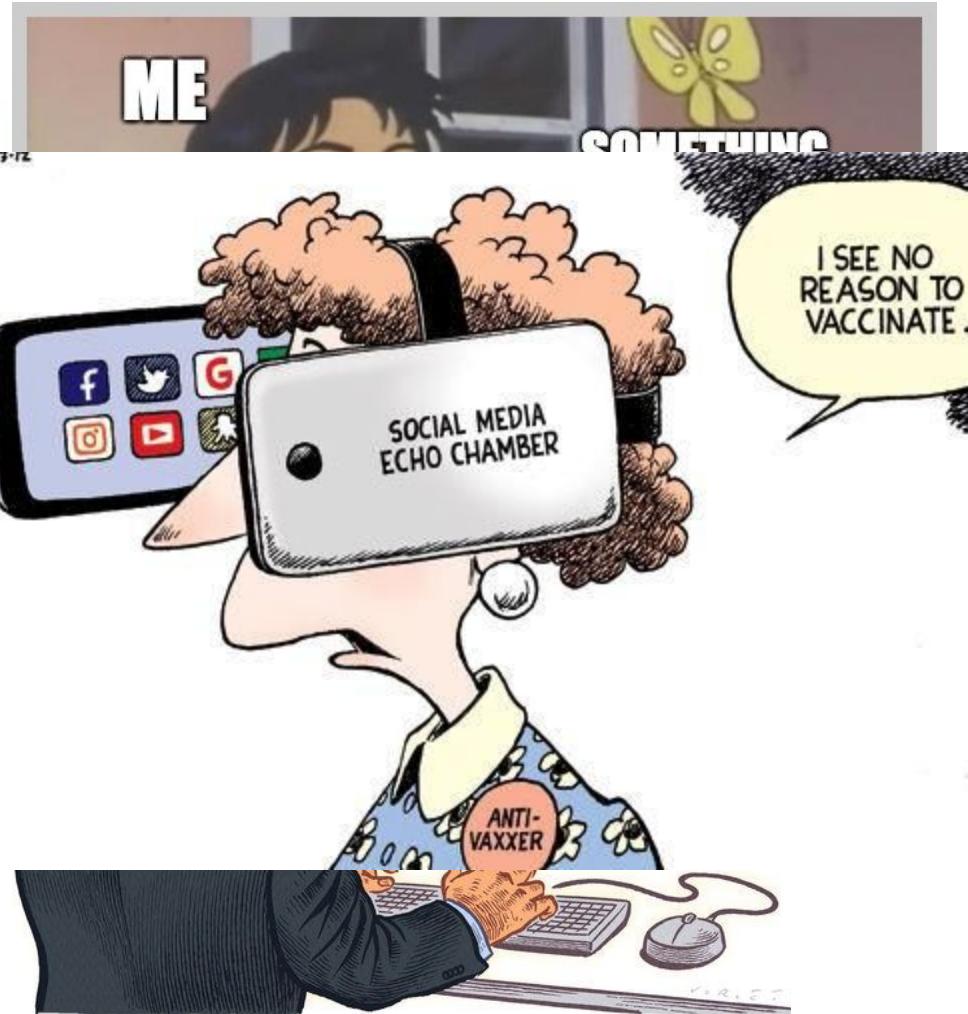
Yo dog, I heard you know recommender system



Can you make a recommender system that  
recommends the right recommender system?

I SPEND MORE TIME LOOKING FOR A  
MOVIE TO WATCH

THAN WATCHING IT



How likely are you to recommend Windows 10 to a friend or colleague?

1

2

3

4

5

Not at all likely

Extremely likely

Please explain why you gave this score.

I need you to understand that people don't have conversations where they randomly recommend operating systems to one another



**MOM**



Page 1 of  
google

Page 2 of  
google

