Matching

Lecture 3: Searching for candidates

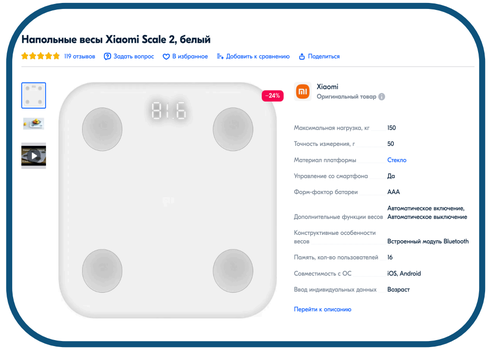
Anton Ryabtsev

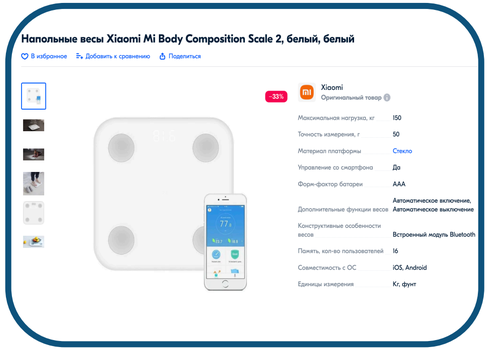
Moscow Institute of Physics and Technology

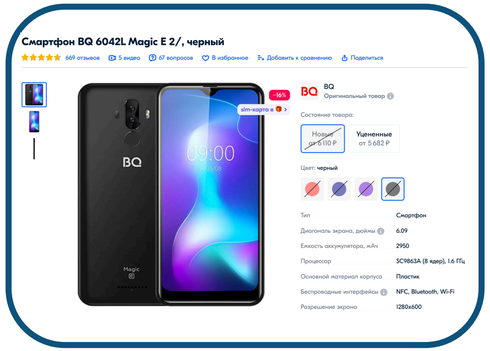
Autumn 2023

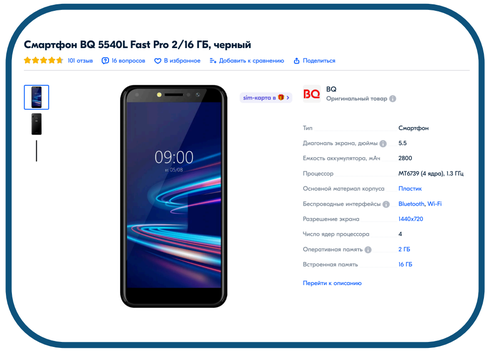
What provides pairs?

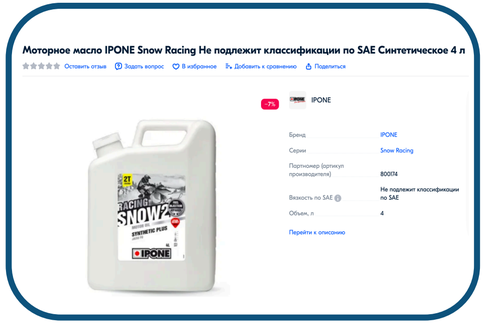
?





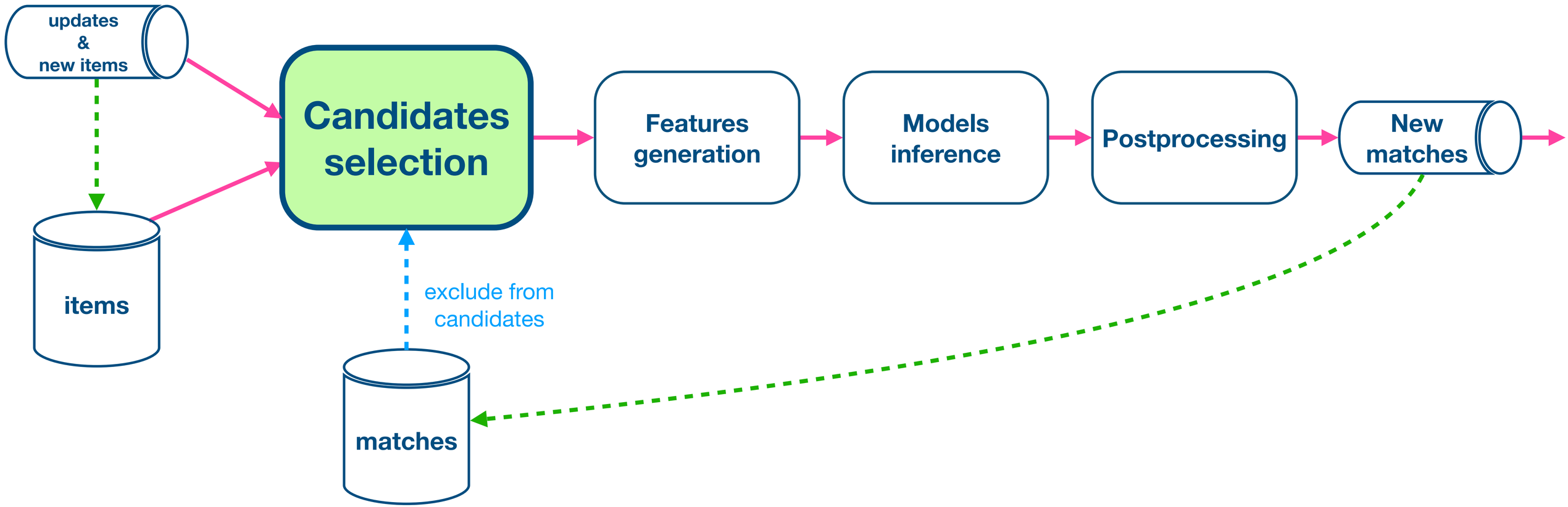






Matching Pipeline

High level design



Narrowing the Search Space

Negatives that are easy to recognize

≠

≠

USB Adapter ≠ **Sweatshirt with print**

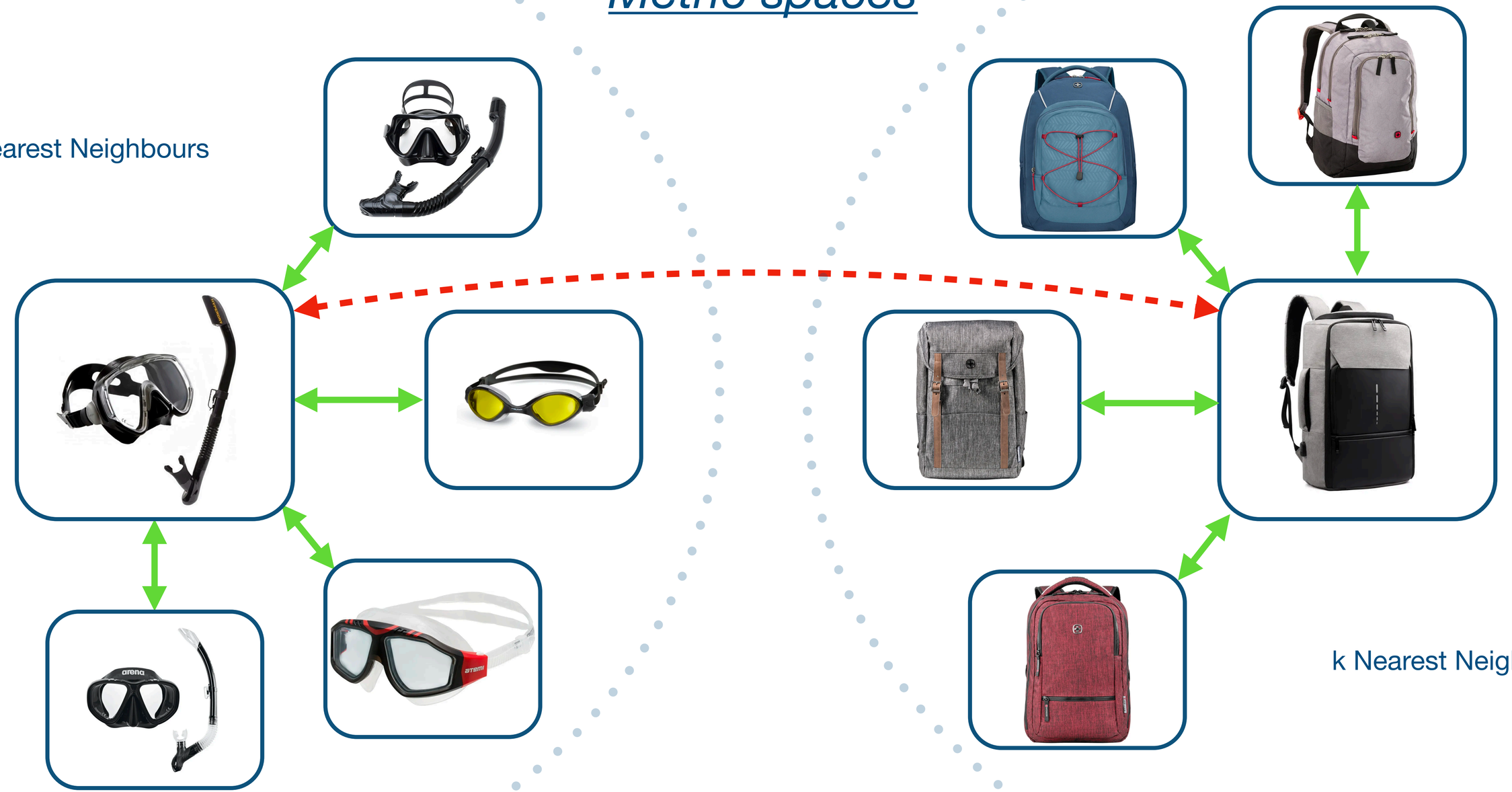
Banana chips ≠ **Cappuccinatore Kitfort**

Outdoor thermometer ≠ **Dried mango «King»**

Narrowing the Search Space

Metric spaces

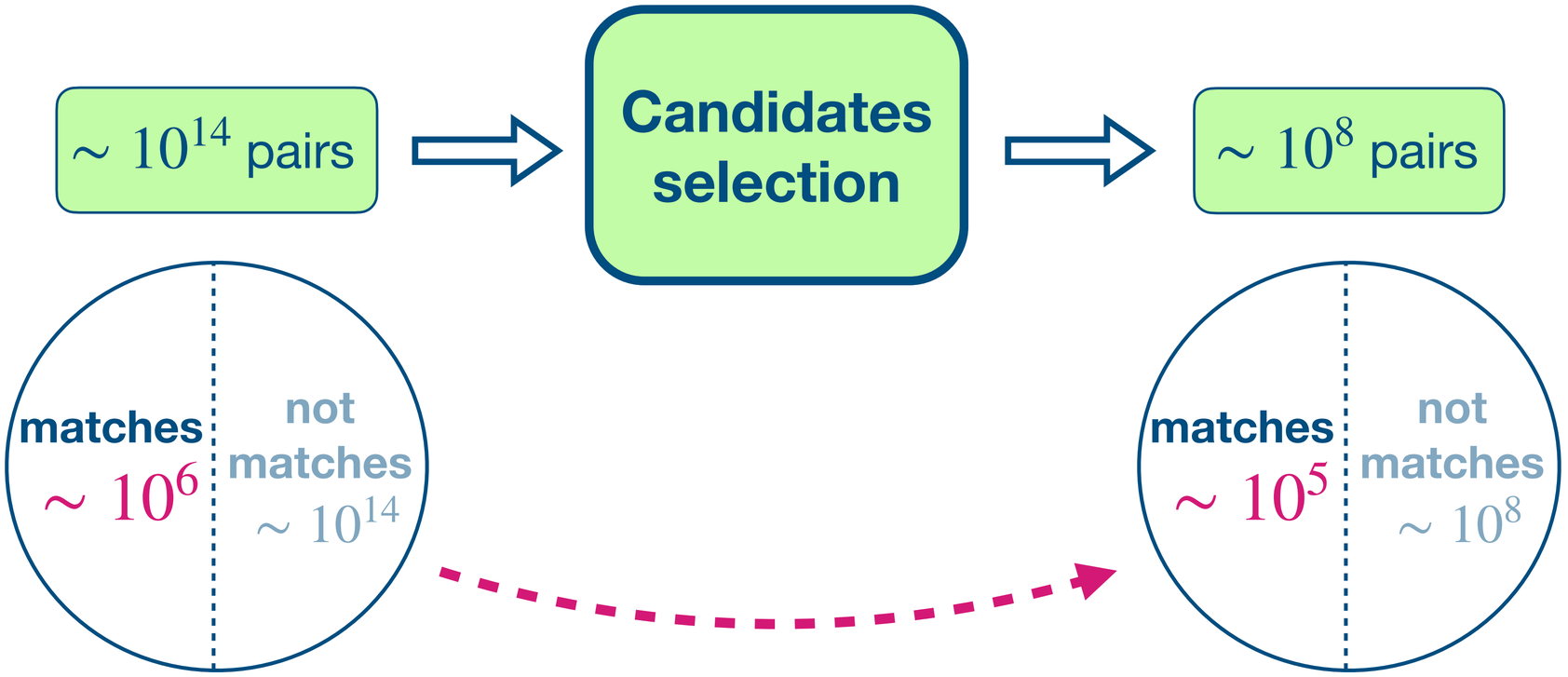
k Nearest Neighbours



k Nearest Neighbours

Quality Evaluation

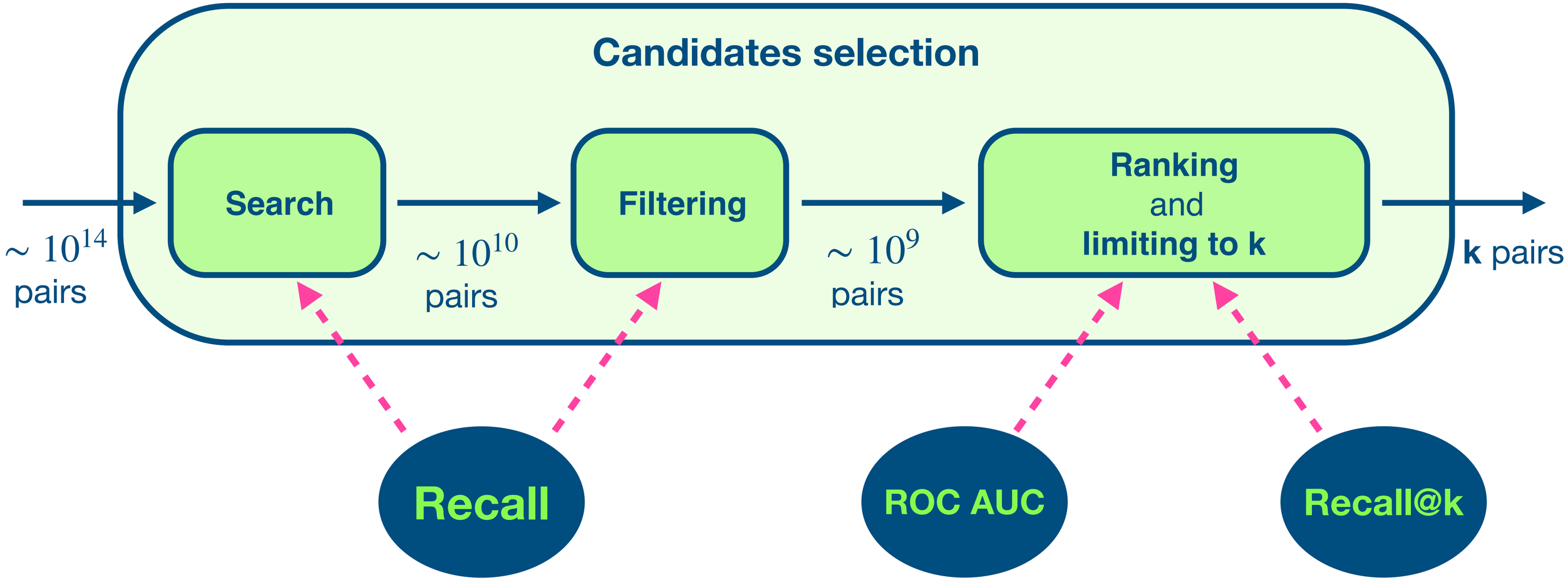
What are we risking?



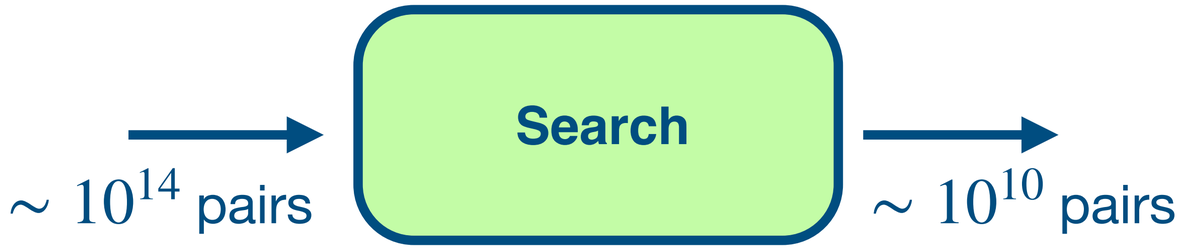
Recall

Candidates selection stages

What's under the hood?



Search



Search

Straightforward approach

(0.25, 0.52, …, 0.18)

(0.22, 0.56, …, 0.12)

(0.21, 0.51, …, 0.19)

(0.24, 0.42, …, 0.89)

(0.23, 0.39, …, 0.92)

(0.20, 0.39, …, 0.92)

(0.18, 0.42, …, 0.17)

(0.12, 0.40, …, 0.13)

(0.13, 0.36, …, 0.90)

(0.20, 0.39, …, 0.92)

Object	Distance
	12
	3
	14
…	…

Search

Straightforward approach

Object	Distance
	12
	3
	14
...	...

Object	Distance
	5
	17
	2
...	...

Object	Distance
	6
	15
	0
...	...

...

Search

Straightforward approach

Pros:

- Exhaustive search — the maximum possible recall is 100%.

Cons:

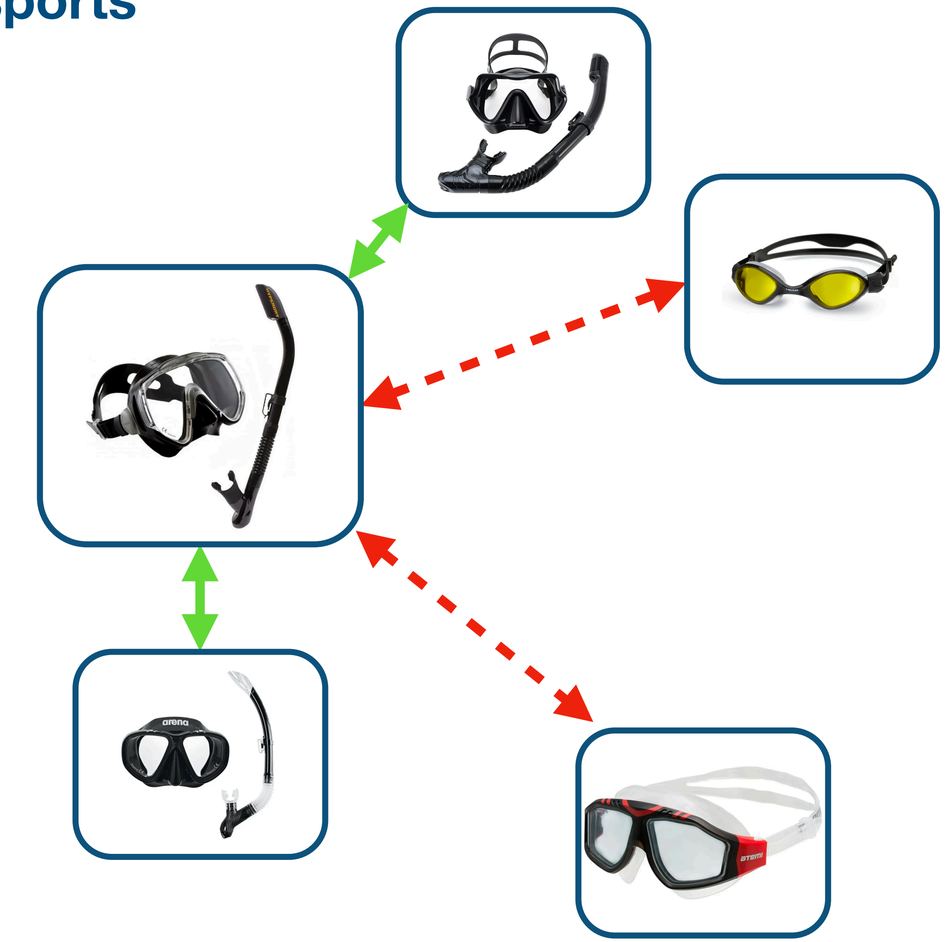
- Exhaustive search — quadratic complexity $O(m \times n)$, $m \sim 10^6$, $n \sim 10^8$ (too long for production).

Search Optimizations

Different categories

Accessories

Water sports



Search Optimizations

Different categories

Pros:

- You can search in each category in parallel, and within each category the search will be faster.
- We probably already have some kind of division into categories.

Cons:

- The object could be assigned to the wrong category. Then we won't be able to find a match for it.
- Still an exhaustive search.

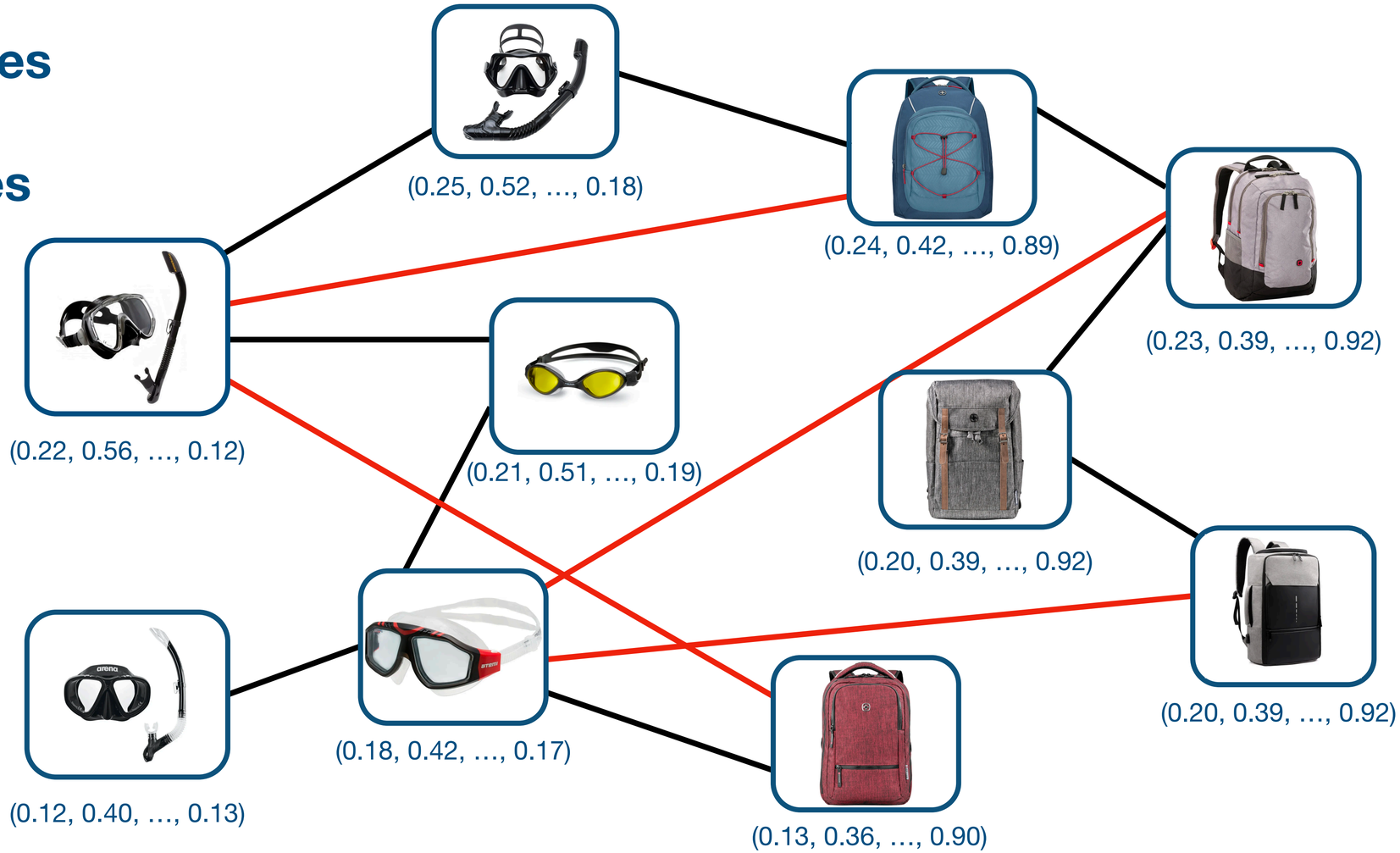
Search Optimizations

Approximate search

- **NSW (Navigable Small World)**

— short edges

— long edges



Search Optimizations

NSW (Navigable Small World)

Graph

- Each point in the graph represents an object in metric space.
- Each edge indicates the possibility of moving from point A to point B.
- The graph is undirected, i.e. the transitions are mutually inverse.
- The edges of the graph must represent both long and short links.

Construction algorithm

1. For each object, distances to random 10% of objects are calculated.
2. From the 100 **closest**, 10 are selected randomly and connected by an edge to the original object.
3. From the 100 **most distant** ones, 10 are selected randomly and connected by an edge to the original object

10%, 100 and 10 — hyperparameters.

Search Optimizations

NSW (Navigable Small World)

(0.25, 0.52, …, 0.18)

(0.24, 0.42, …, 0.89)

(0.23, 0.39, …, 0.92)

(0.22, 0.56, …, 0.12)

(0.21, 0.51, …, 0.19)

(0.20, 0.39, …, 0.92)

(0.20, 0.48, …, 0.08)

(0.12, 0.40, …, 0.13)

(0.18, 0.42, …, 0.17)

(0.13, 0.36, …, 0.90)

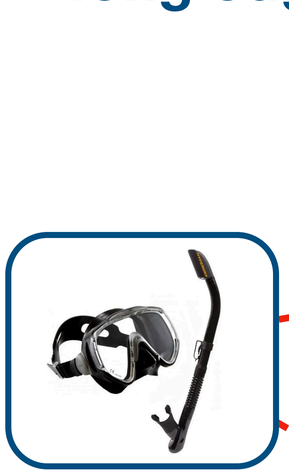
(0.20, 0.39, …, 0.92)

Entry point

Search Optimizations

*HNSW (**Hierarchical** Navigable Small World)*

Hierarchical — long edges first.



(0.22, 0.56, …, 0.12)

(0.24, 0.42, …, 0.89)

(0.23, 0.39, …, 0.92)

(0.20, 0.48, …, 0.08)

(0.18, 0.42, …, 0.17)

(0.13, 0.36, …, 0.90)

(0.20, 0.39, …, 0.92)

Entry point

Search Optimizations

*HNSW (**Hierarchical** Navigable Small World)*

Hierarchical — long edges first.


(0.25, 0.52, …, 0.18)


(0.24, 0.42, …, 0.89)


(0.23, 0.39, …, 0.92)


(0.22, 0.56, …, 0.12)


(0.21, 0.51, …, 0.19)


(0.20, 0.39, …, 0.92)


(0.20, 0.48, …, 0.08)


(0.18, 0.42, …, 0.17)


(0.20, 0.39, …, 0.92)

Entry point


(0.12, 0.40, …, 0.13)


(0.13, 0.36, …, 0.90)

Filtering

Filtering

1. **Heuristic filters by attributes:**
   - Color.
   - Size, dimensions.
   - Pieces.
   - Price.
   - Etc.

Pros:
- Forcibly bans something that should not be matched.

Cons:
- Has a high false positive rate.

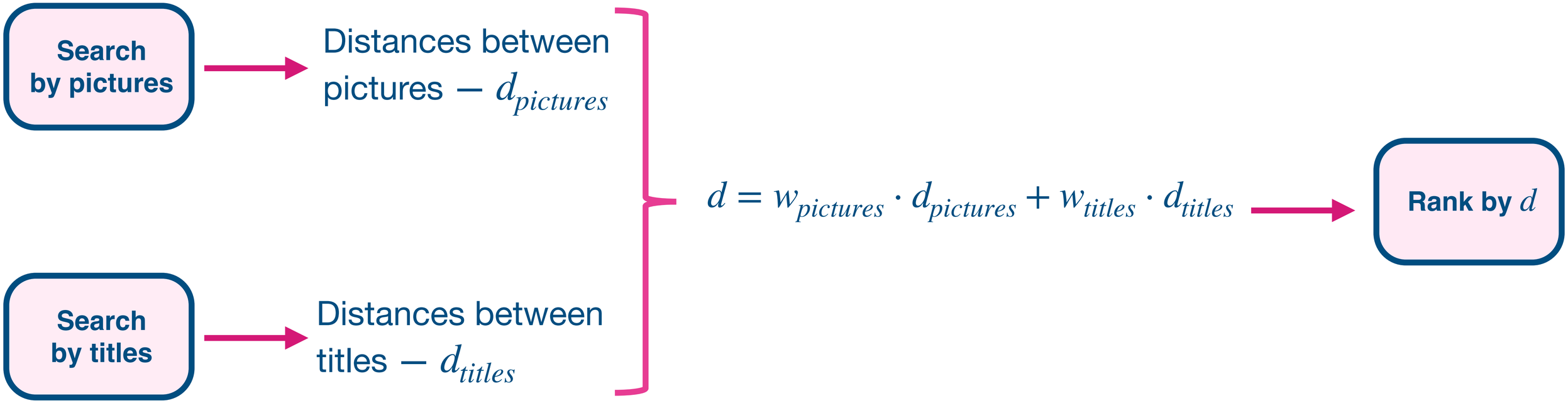
2. **Heuristic filters by distances:**
   - Distance between pictures.
   - Distance between titles.

Pros:
- Bans candidates whose likelihood of being a match is extremely low.

Cons:
- We will not be able to match products without pictures (is this really a disadvantage?).

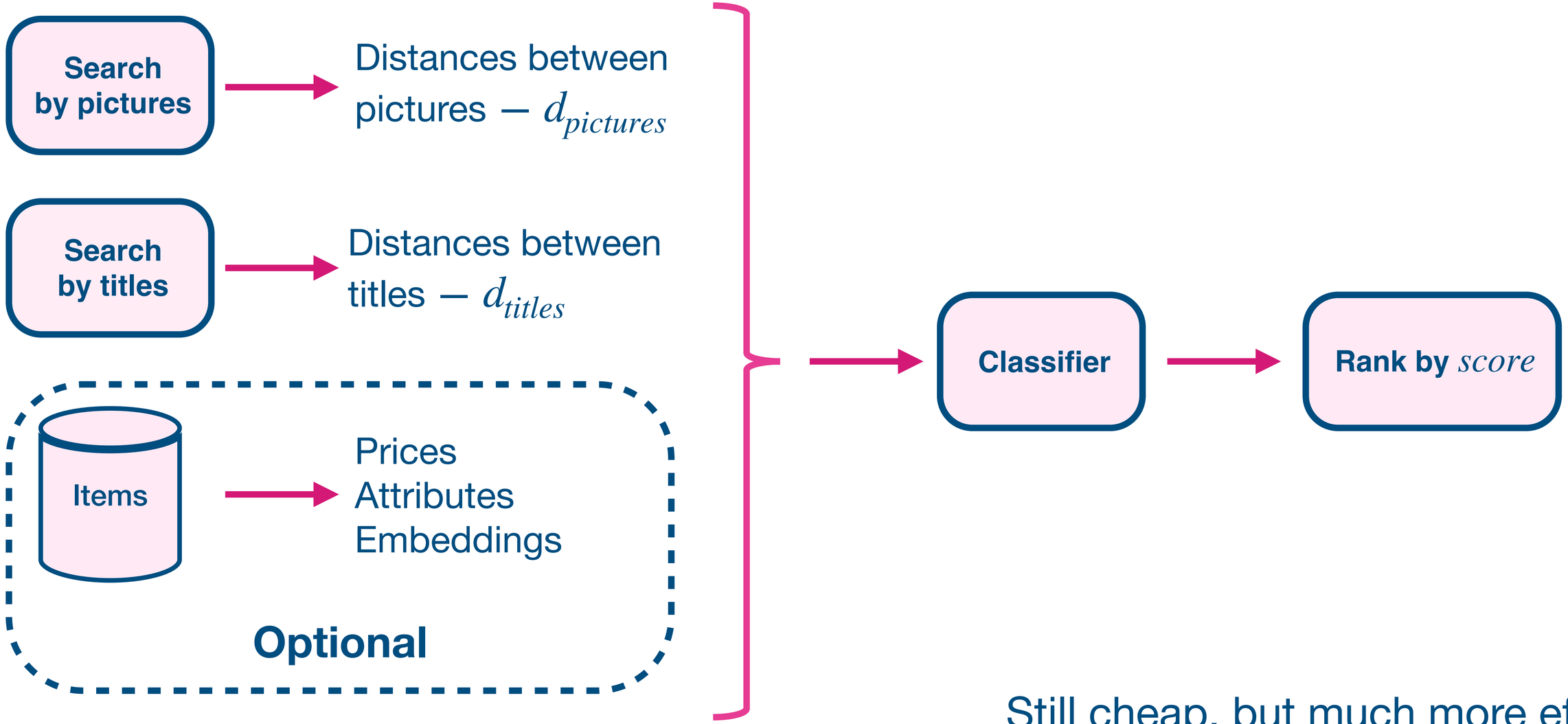
Ranking

Ranking



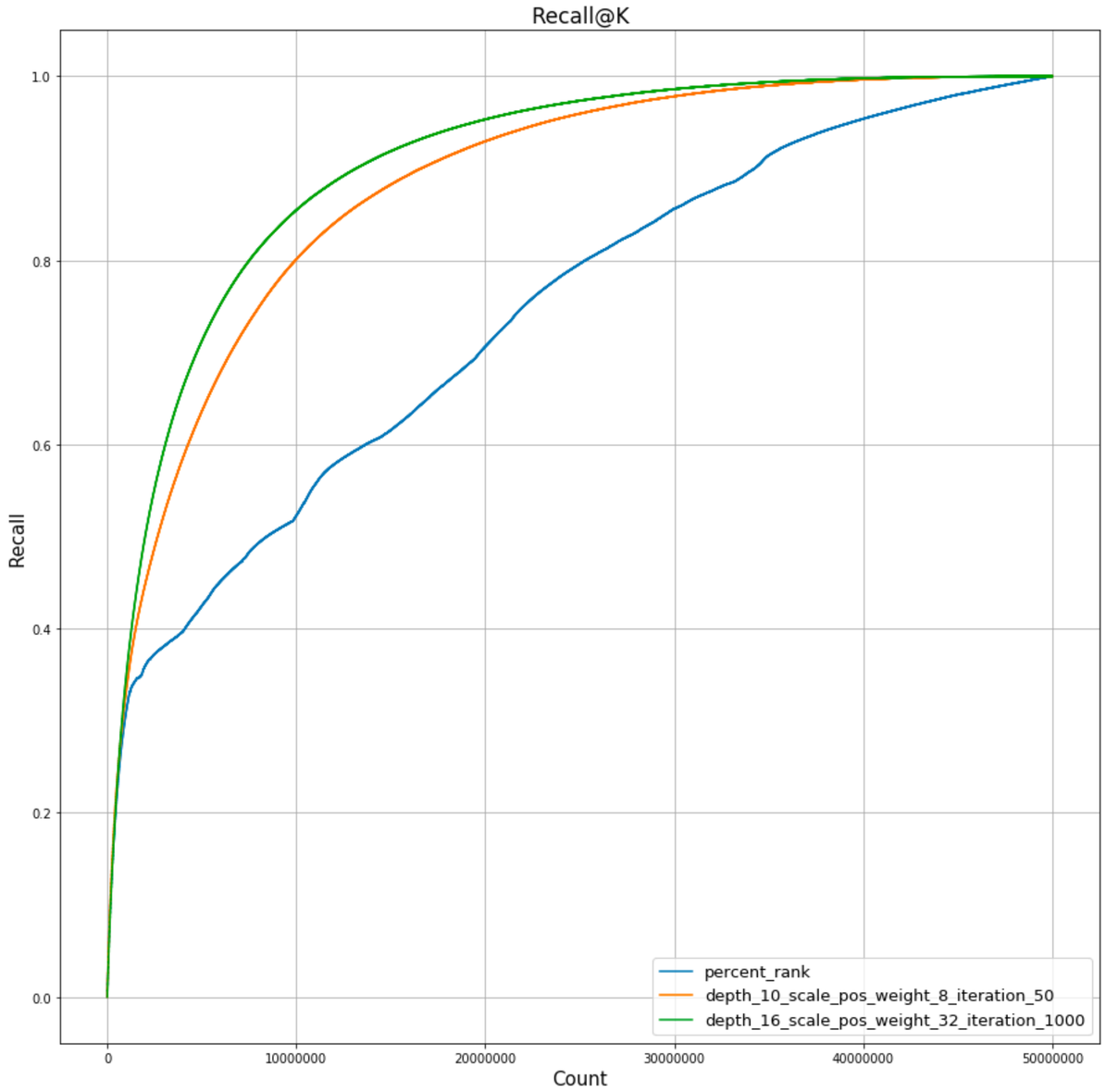
Cheap and cheerful

Ranking



Still cheap, but much more effective

Ranking


Recall@K
Recall
Count
1.0
0.8
0.6
0.4
0.2
0.0
0
10000000
20000000
30000000
40000000
50000000
percent_rank
depth_10_scale_pos_weight_8_iteration_50
depth_16_scale_pos_weight_32_iteration_1000

Important to keep in mind

Secrets, tricks and folk wisdom

- Each optimization leads to a decrease in **recall**. You need to find the optimal balance between **speed** and **recall** and constantly ensure that the balance remains optimal.
- Filters are a crutch. It’s better to make features based on them - the model will be able to process this information smarter.
- It is imperative to track how much **recall** is lost at each step.