elastic

tinder

CUSTOMER STORY: SEARCH

Software & technology · Online dating · Recommendation retrieval

Tinder drives real-time recommendations across 190 countries with 4x lower CPU on Elasticsearch

Tinder® built its large-scale recommendation architecture on open source Elasticsearch, delivering personalized matches to millions of users across 190 countries and 45 languages with 4x lower CPU usage.

4x
Lower CPU usage, geo-indexed retrieval

47%
Lower scoring latency in ranking

60%
Higher filtering performance

190
Countries, 45 languages, real time

The challenge

- ✕ Every recommendation must surface from tens of millions of profiles across 190 countries and 45 languages, at extremely low latency.
- ✕ Hard filters like age, gender, and distance alone still leave hundreds of thousands of candidates in dense areas.
- ✕ Retrieval efficiency sets the speed, cost, and quality of every downstream ranking step.
- ✕ The system has to keep adapting to changing user behavior while holding infrastructure cost down.

The solution

- ✓ Elasticsearch on Amazon EKS powers retrieval, narrowing tens of millions of profiles to roughly a thousand candidates.
- ✓ Geo-indexing partitions users so Elasticsearch scans only nearby profiles, cutting CPU and query latency.
- ✓ A custom Roaring bitmap filter compresses each user's exclusion list, up to ~70,000 actioned profiles, into one object.
- ✓ A custom plugin runs LightGBM scoring compiled to Java inside Elasticsearch, on compact binary feature sets.

How it works

1. Retrieve
Elasticsearch narrows tens of millions to ~1,000

2. Score in-engine
Java/LightGBM scoring runs inside Elasticsearch

3. Rank
GPU deep-learning models weigh richer signals

4. Rerank & deliver
Balance results, send to the app

Before

- Hard filters alone still left hundreds of thousands of candidates in dense areas
- Features fetched per candidate separately, adding latency
- CPU cost and latency scaled with the global user base

After

- → Retrieval narrows to ~1,000 candidates before costly ranking runs
- → Geo-indexed search scans only nearby profiles: 4x lower CPU
- → Roaring bitmap and binary features cut filtering and extraction time

> **"If we do a really good job in retrieval, we don't actually need to run ranking for 10,000 candidates. By doing that, we can save quite significant cost for ranking infrastructure. And that creates a natural pressure on the quality of our retrieval."**
>
> — Igor Sokolov, Staff Software Engineer, Tinder

Why it matters beyond matchmaking

Efficient retrieval is what makes the rest affordable: by narrowing to a small, high-quality candidate set inside Elasticsearch, Tinder avoids ranking tens of thousands of profiles downstream and keeps infrastructure cost in check. The same architecture is a flexible foundation for what comes next, from newer vector search to additional ML models.

See how Elasticsearch powers retrieval and ranking at scale or start a free trial →

