1、kd树原理介绍：

1-1、构建：kd是二叉搜索树的一种。

1-2、构建方式：每个节点按照某一个维度
进行切分，按照
维度的选择的方式一般
是先选方差的大小，
选方差大的维度进行
对数据的切分，因为区分度大。

1、向量化检索

2、YouTube 召回、YouTube Rank

3、模型部署常用方法。○

1、隐式召回/embedding 召回。

流程：离线训练好 vector(item)　facebook | C++

↓

将这些 vector 存入某结构 ⇒ faiss

↓

user 请求 → get user_emb

↓

v → 检索 v

最近邻的 top N.

某结构：faiss、kd-tree、Annoy、HNSW.

kd-tree → vector 的维度较低时，不超过 30 维度。

1、id：800 万个，dim：200.
faiss：30ms 以内。

向量化检索的流程：

1、离线训练（i2v、双塔）.

2、双塔已经有了，可推荐的 item → get_id → get_特征 → 过 item 侧的塔 → get item 的 embedding. → 放入 faiss 做 train.

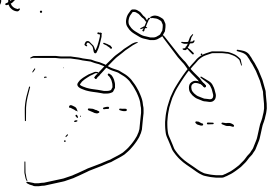
3、user 请求 → get_user_embedding（过 user 侧塔）→ faiss 检索。→ 业务规则的过滤。} serving

1. KNN、kd-tree.

kd-tree: 二叉树. 二叉搜索树.

[2, 3]. → kd-tree

float: 2.3 → 查找.



$\log(N)$.

[1, 2, 3, 4, 5]

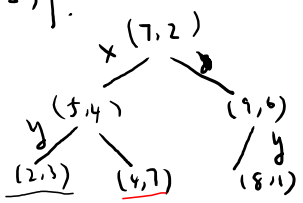
[2, 3, 3, 4, 5]

[2, 3, 3, 4, 5]

[1, 2, 3, 4, 5]

1
2
3 对比心?
4
5

eg. [(2,3), (5,4), (9,6), (4,7), (8,1), (7,2)].



(2.1, 2.8)

(2, 4.5).