DEV DAY

Vector Search: Beginner to Pro

WiFi Password: loveyourdevelopers

Hands-on session

- There are no stupid questions
- Actively discuss
- Be respectful

MongoDB Skill Badges

Data Modeling

- MongoDB Skill — Relational to Document Model
- MongoDB Skill — Schema Patterns and Antipatterns
- MongoDB Skill — Advanced Schema Patterns and Antipatterns
- MongoDB Skill — Schema Design Optimization

Monitoring/Tuning/Automation

- MongoDB Skill — Monitoring Tooling
- MongoDB Skill — Performance Tools & Techniques

Performance at Scale

- MongoDB Skill — Cluster Reliability

Gen AI

- MongoDB Skill — Vector Search Fundamentals
- MongoDB Skill — RAG with MongoDB
- MongoDB Skill — AI Agents with MongoDB

Security

- MongoDB Skill — Secure MongoDB Self-Managed: AuthN & AuthZ
- MongoDB Skill — Secure MongoDB Atlas: AuthN and AuthZ
- MongoDB Skill — Networking Security: Self-Managed
- MongoDB Skill — Networking Security: Atlas

Query

- MongoDB Skill — Query Optimization
- MongoDB Skill — CRUD Operations

Aggregation

- MongoDB Skill — Fundamentals of Data Transformation

Sharding

- MongoDB Skill — Sharding Strategies

Indexes

- MongoDB Skill — Indexing Design Fundamentals

Architecture

- MongoDB Skill — MongoDB Overview

Search

- MongoDB Skill — Search Fundamentals

Skill Badge Benefits

Formal Validation

Provide a way to quickly learn and validate specific MongoDB skills

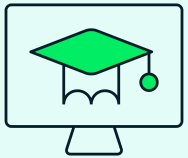
Digital Badges

Showcase achievements with a Credly badge and inclusion in the Talent Directory.

Elevate in current roles and increase appeal for future positions.

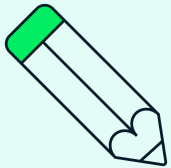
Career Advancement

Free, Focused Credentials



Learn

Learn and practice during this workshop.



Validate

Showcase your knowledge in a short 10-question skill check.

Earn

Claim your Credly badge and share it on social media.

Earn a badge in 60 – 90 minutes.

Agenda

- What is vector search?
- What are embeddings?
- Vector search in MongoDB
- Tuning vector search queries
- Q&A

Lab instructions

👐, 	Hands-on sections
<CODE_BLOCK_N>	Code placeholders
📚	Reference documentation for the code blocks in the cell that follows

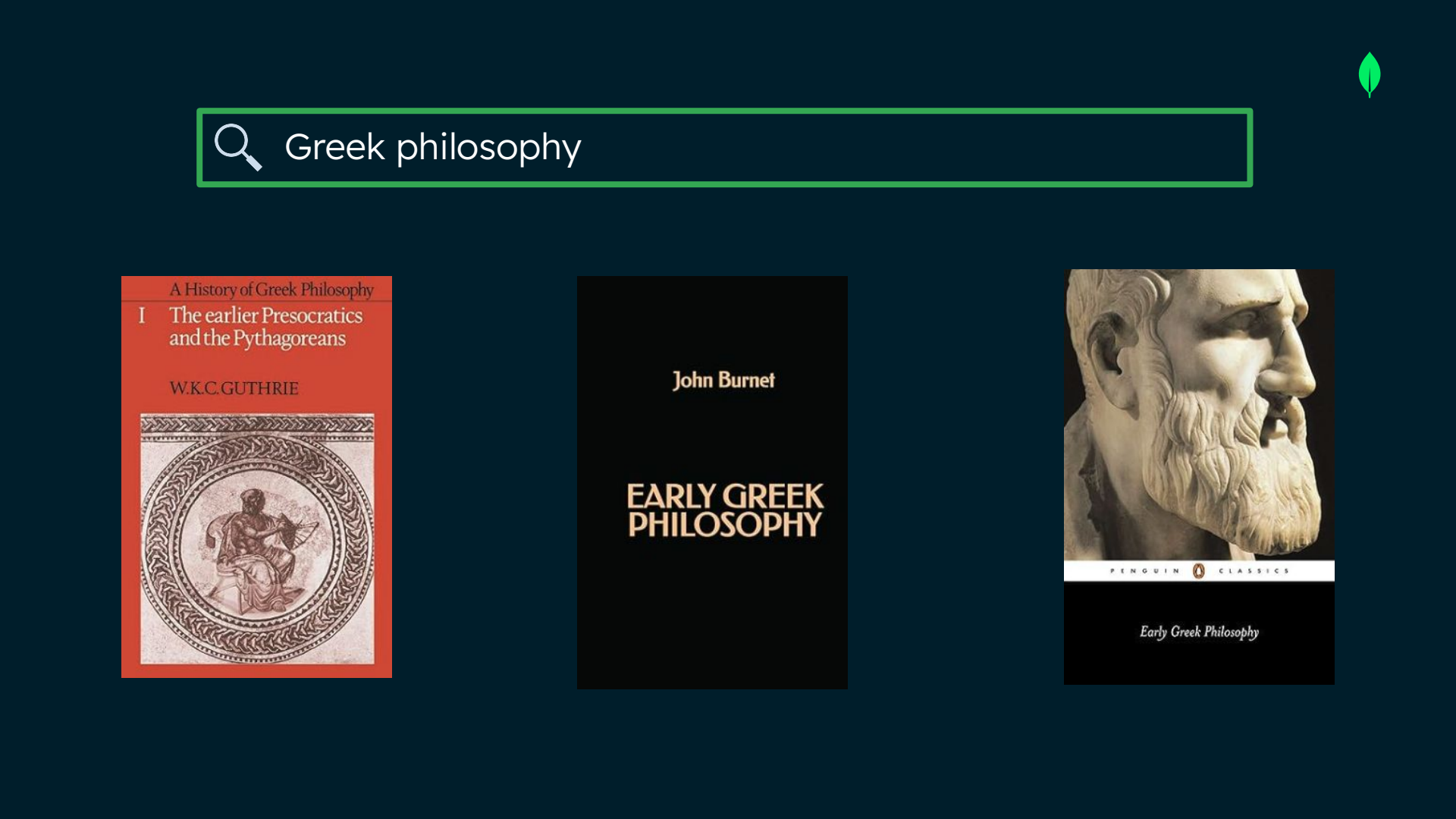
Lab setup

PASSKEY = “replace-me”

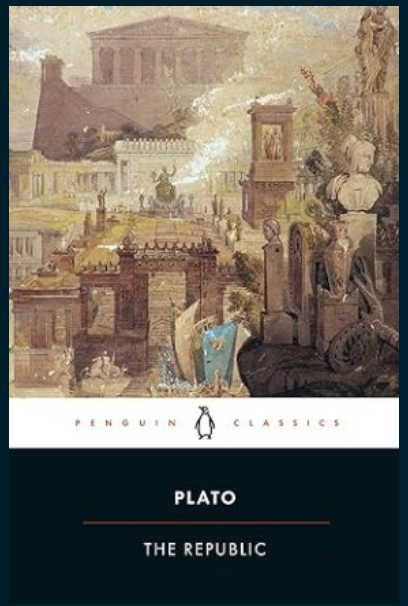
TODO:

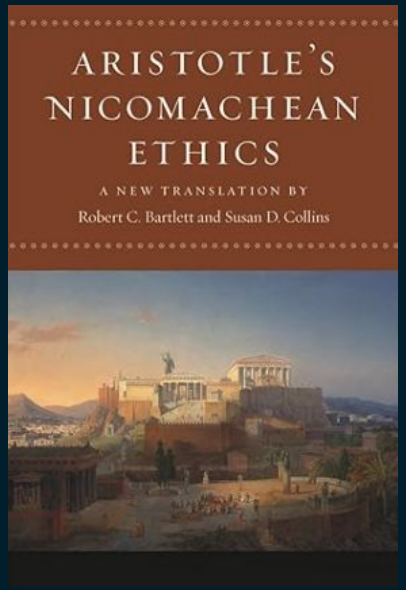
- Navigate to the lab: mdb.link/devrel-vector-search
- Navigate to the lab setup: mdb.link/lab-setup
- Complete the 👐 modules

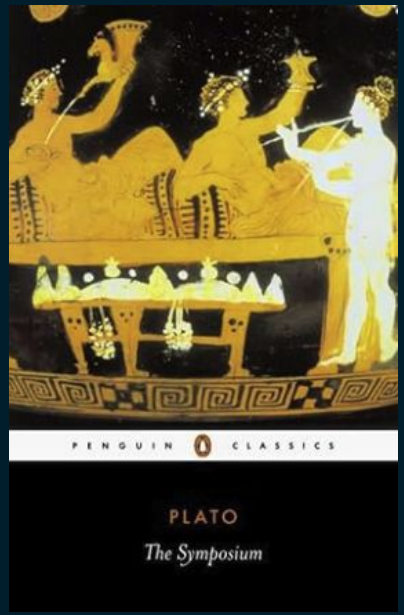
Lexical vs. Vector Search


Greek philosophy
A History of Greek Philosophy
I The earlier Presocratics and the Pythagoreans
W.K.C. GUTHRIE
John Burnet
EARLY GREEK PHILOSOPHY
PENGUIN CLASSICS
Early Greek Philosophy

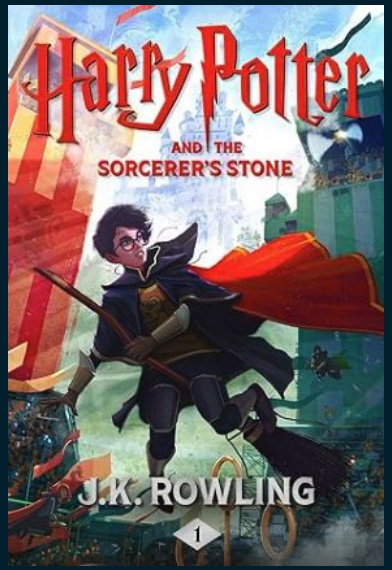
Greek philosophy

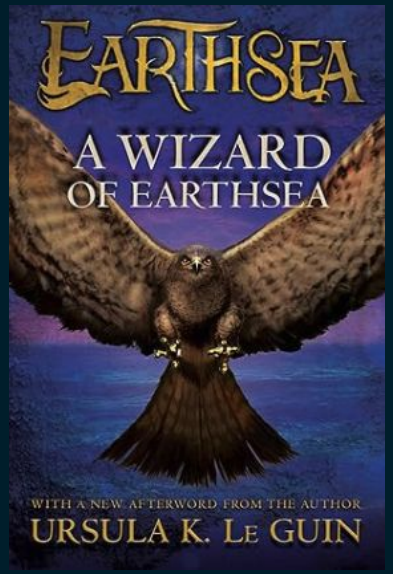


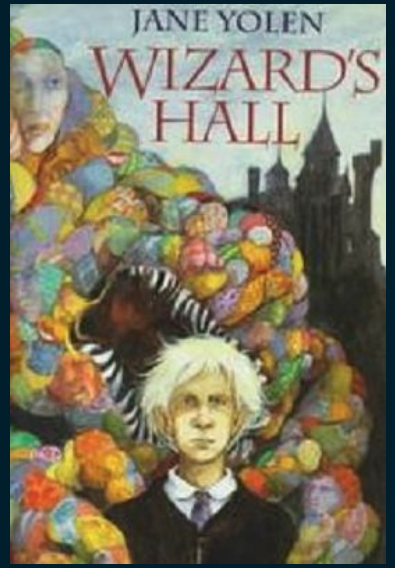




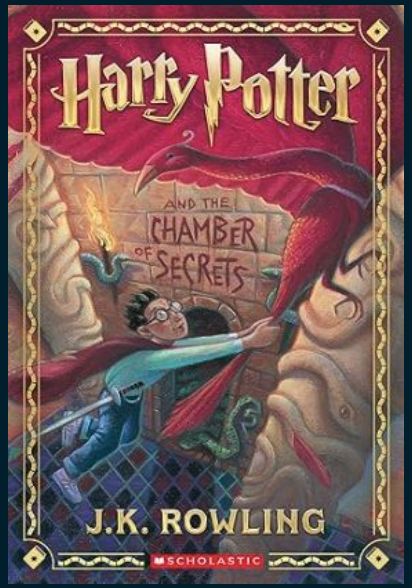
Fantasy novel about a boy in a school of magic

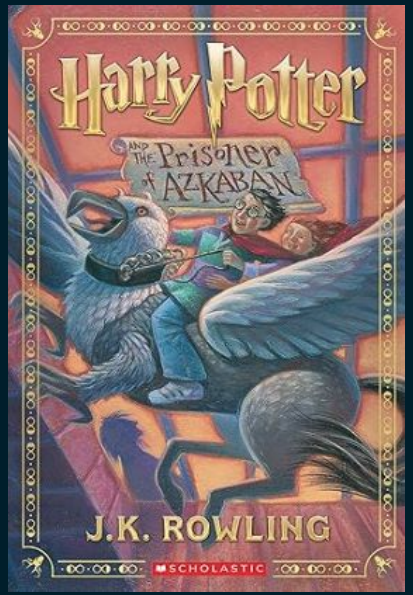


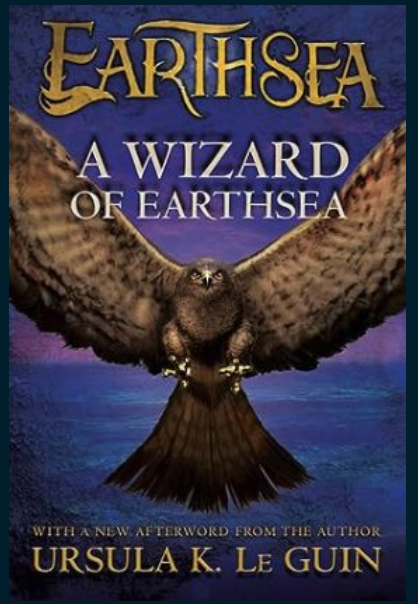




harry_potter_1_book_cover.png







Lexical search

What?

- Keyword search

When?

- Your text corpus closely matches how users search
- First pass at text-based relevancy

Vector search

What?

- Semantic similarities

When?

- “Vocabulary gap” between corpus and how users search
- Text, image, audio, video search

Glossary

Embedding

Representation into vectors of the semantic meaning produced by LLM

Embedding Model

AI model that understands the human language meaning of large input

Vector and Dimensions

A numerical array representing a point in a multidimensional space, often used as an angle to the axis, and a length

Vector Indexing

An inverted index on vectors to find and score embedding with similarity

Vectors

Vectors = numerical representations of unstructured data.

- encoded as an array
- each element represents a dimension
- generated using embedding models

Vectors example: store

SKILL

[aisle, bay]

[aisle, bay]

[aisle, bay, brand, color]

Thousands of dimensions 🤯

Embeddings

Converting text to vectors

If you're new to machine learning, you might wonder how plain text gets transformed into vectors — mathematical objects that power semantic search.

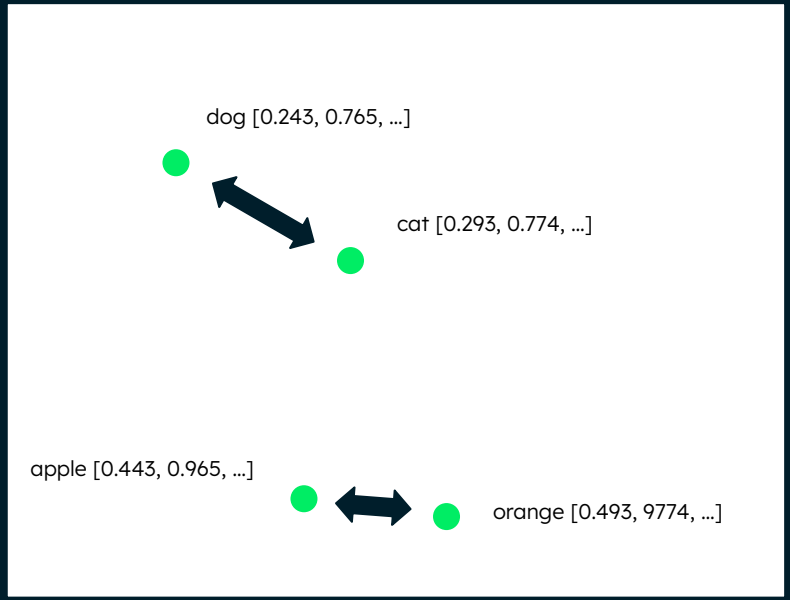
Let us build some intuition with a simple example using character bigrams.

“Aircraft” -> "ai","ir","rc","cr","ra","af","ft"

How many possible bigrams of alphabets (Dimensions) - 26*26= 676

Vector representation of “Aircraft” : { "ai":1,"ir”:1,"rc":1,"cr":1,"ra":1, "af":1,"ft":1,"aa": 0, "ab":0, ...}

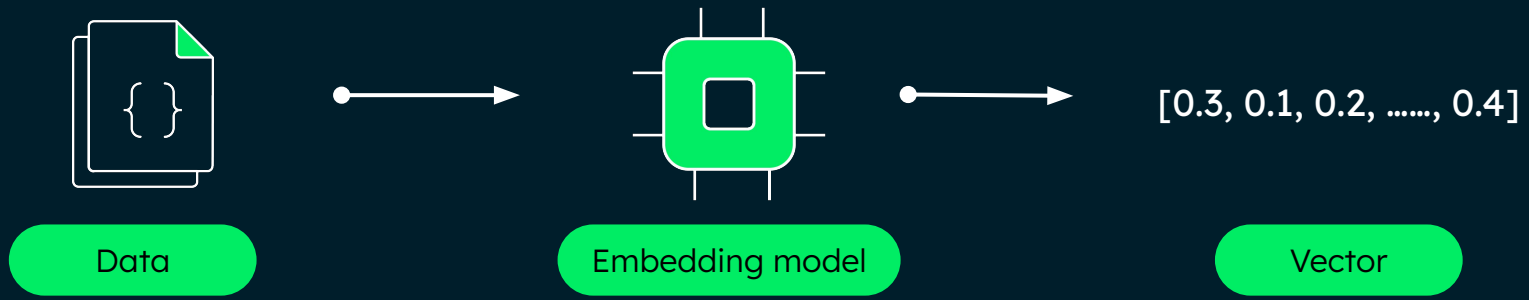
Embeddings



Numeric, multi-dimensional representation of a piece of information

- Capture semantic qualities of data
- Semantically similar data ends up close together in vector space

How to embed data



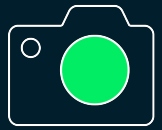
Text

Voyage AI

Multimodal Embedding model

[0.3, 0.1, 0.2, ..., 0.4]

Vector



Image

Voyage AI

Multimodal Embedding model

[0.3, 0.1, 0.2, ..., 0.4]

Vector

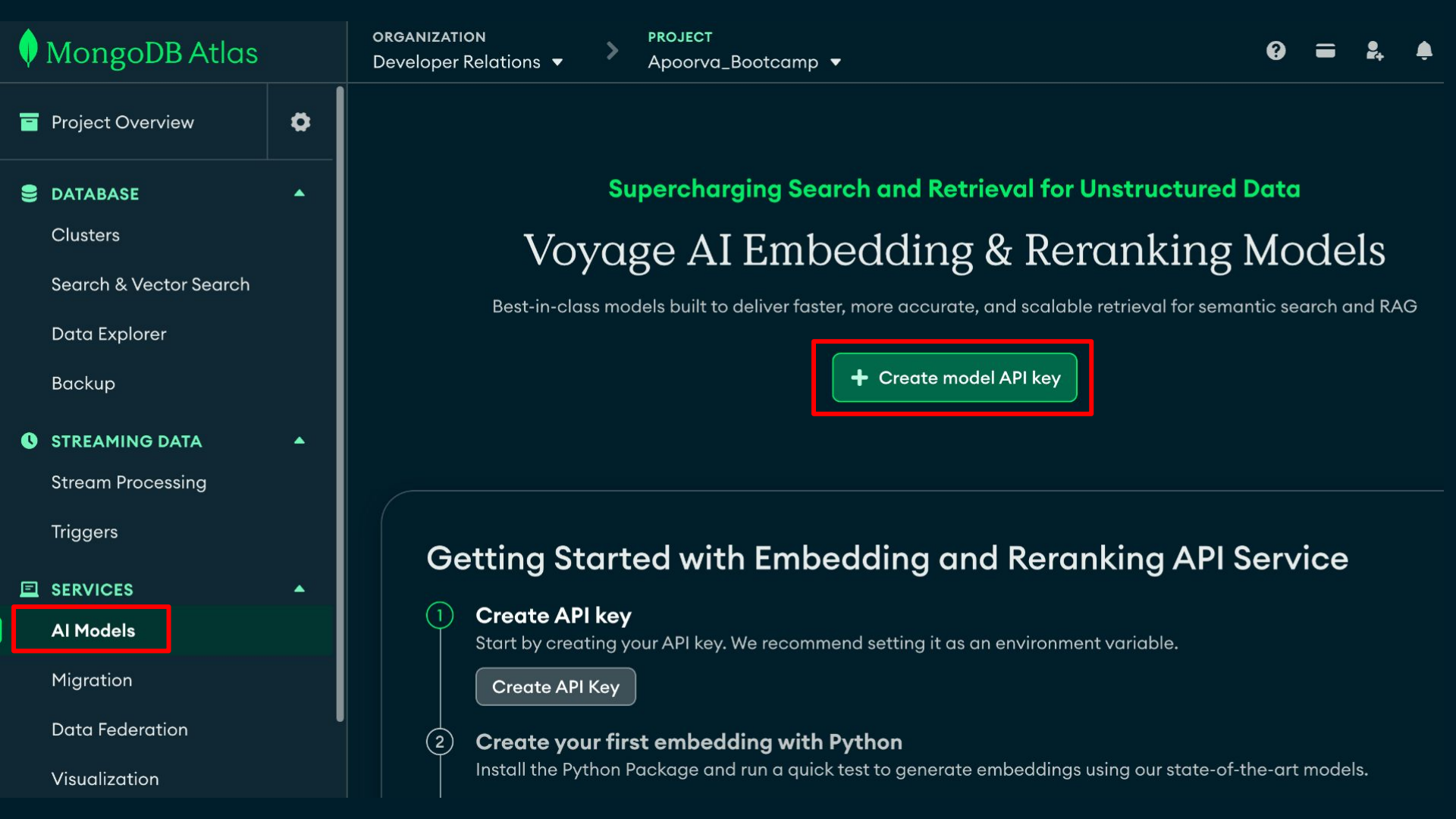
Video

Voyage AI

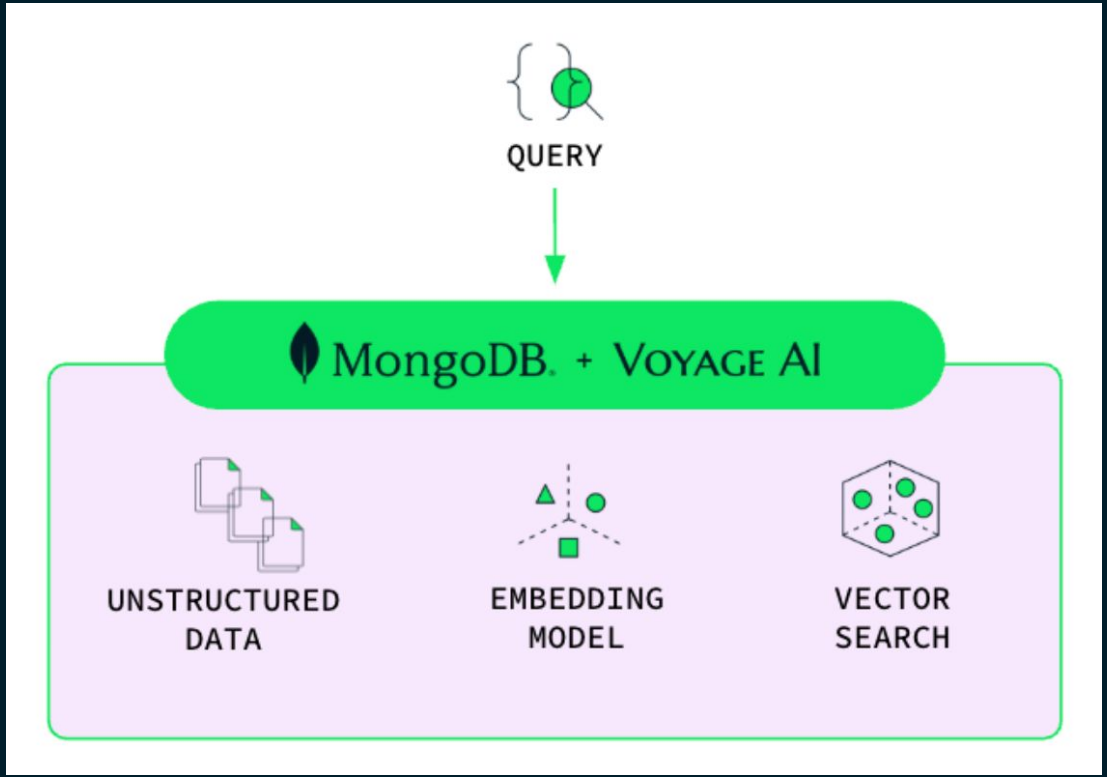
Multimodal Embedding model

[0.3, 0.1, 0.2, ..., 0.4]

Vector


MongoDB Atlas
ORGANIZATION
Developer Relations
PROJECT
Apoorva_Bootcamp
Project Overview
DATABASE
Clusters
Search & Vector Search
Data Explorer
Backup
STREAMING DATA
Stream Processing
Triggers
SERVICES
AI Models
Migration
Data Federation
Visualization
Supercharging Search and Retrieval for Unstructured Data
Voyage AI Embedding & Reranking Models
Best-in-class models built to deliver faster, more accurate, and scalable retrieval for semantic search and RAG
+ Create model API key
Getting Started with Embedding and Reranking API Service
1
Create API key
Start by creating your API key. We recommend setting it as an environment variable.
Create API Key
2
Create your first embedding with Python
Install the Python Package and run a quick test to generate embeddings using our state-of-the-art models.

Auto-embedding in MongoDB Atlas Vector Search

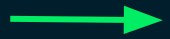


How to choose an embedding model

mdb.link/embedding-models

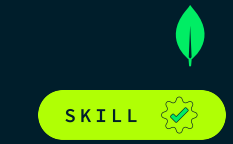
Adding embeddings to existing data

```
_id: "0028608488"
title: "David Copperfield's Tales of the Impossible"
cover: "https://images.isbndb.com/covers/22/86/9780061052286.jpg"
year: 1995
pages: 385
synopsis: “David Copperfield, Arguably The Greatest Illusionist-magician…”
```



```
_id: "0028608488"
title: "David Copperfield's Tales of the Impossible"
cover: "https://images.isbndb.com/covers/22/86/9780061052286.jpg"
year: 1995
pages: 385
synopsis: “David Copperfield, Arguably The Greatest Illusionist-magician…”
embedding: Array
   0: 0.03898080065846443
   1: -0.05879044905304909
   2: 0.04323239979442215
   .
   .
   512: 0.034234036451233547
```

Recap



- Embeddings are an array of numbers that capture semantic attributes of data.
- Embeddings are generated by specialized ML models.
- Different embedding models produce embeddings of different sizes.
- Embeddings can be added in-place into existing MongoDB documents.

Lab setup

PASSKEY = “+8wWcA”

TODO:

- Navigate to the lab: mdb.link/devrel-vector-search
- Navigate to the lab setup: mdb.link/lab-setup
- Complete the 👐 modules

In this exercise, you will:

- Import a dataset into MongoDB.
- Add embeddings to your MongoDB data.

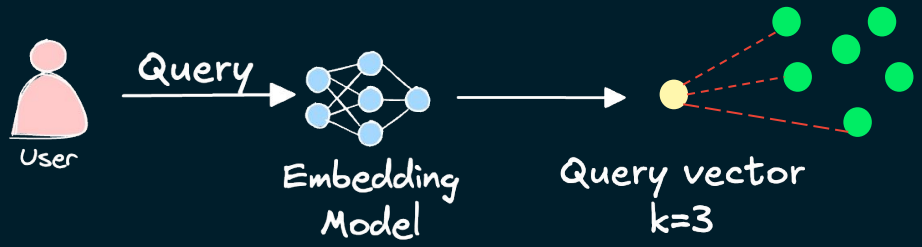
TODO:

- Navigate to mdb.link/devrel-vector-search
- Complete **Step 2-4** in the notebook

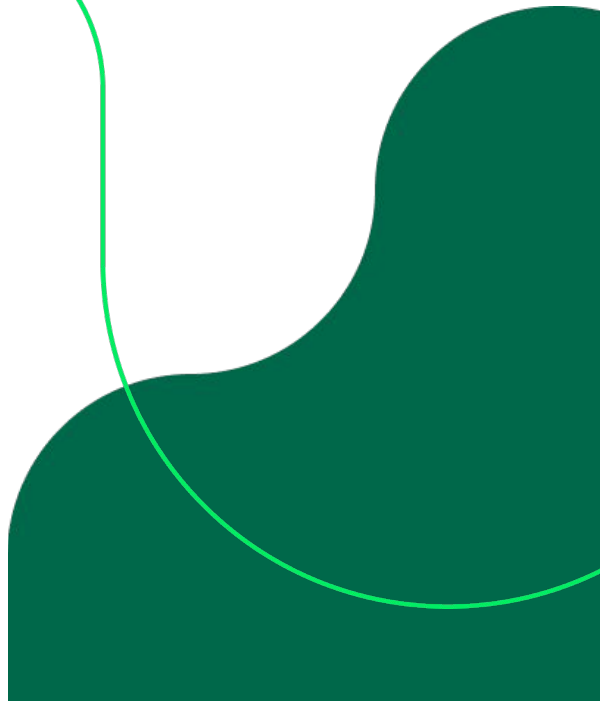
Vector Search

Vector search

Search based on intent/meaning using embeddings

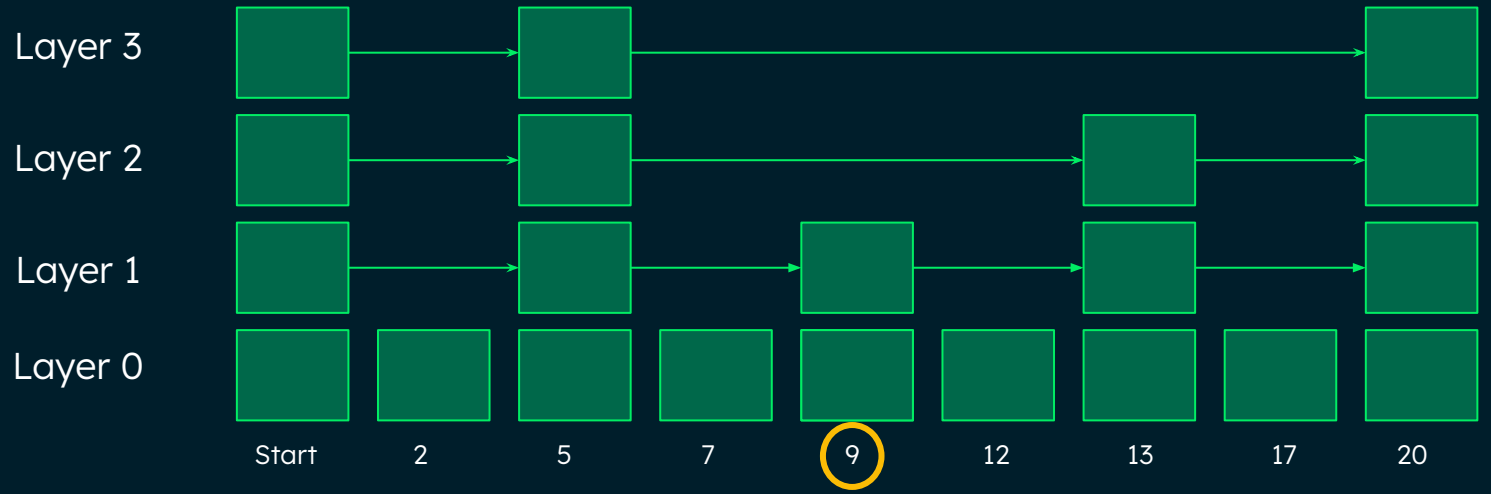


Indexing Algorithms



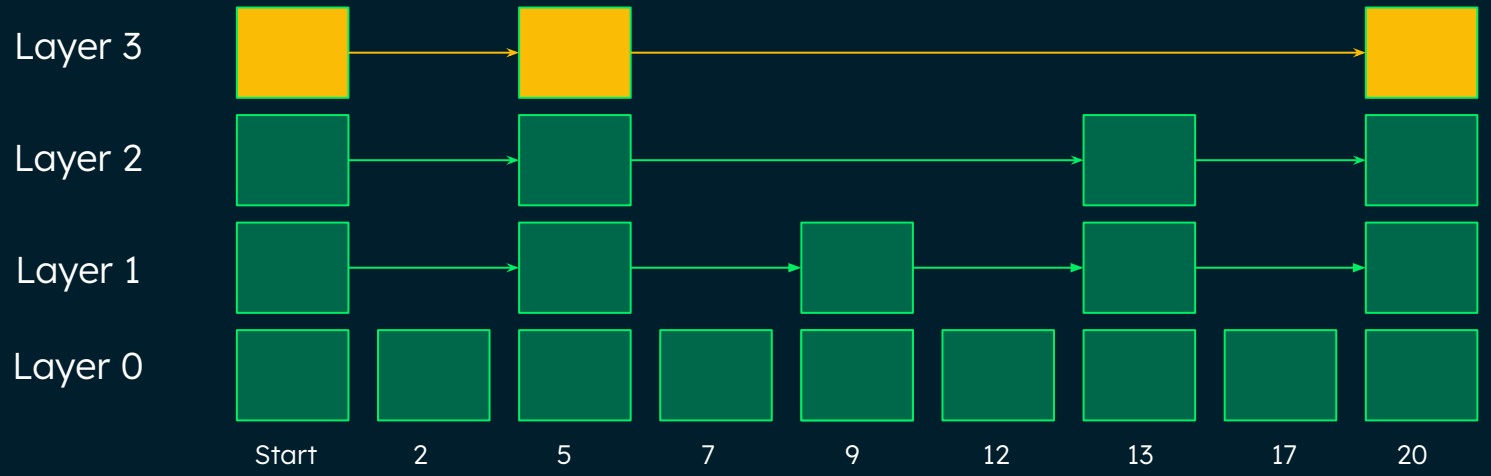
Skip List

Find 9



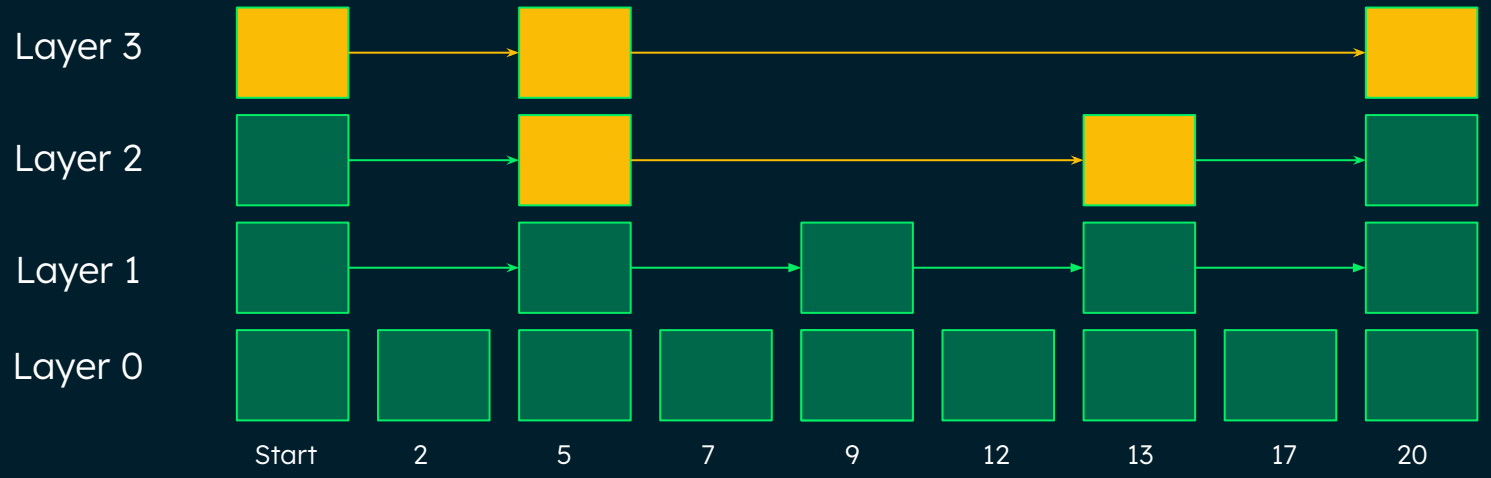
Skip List

Find 9



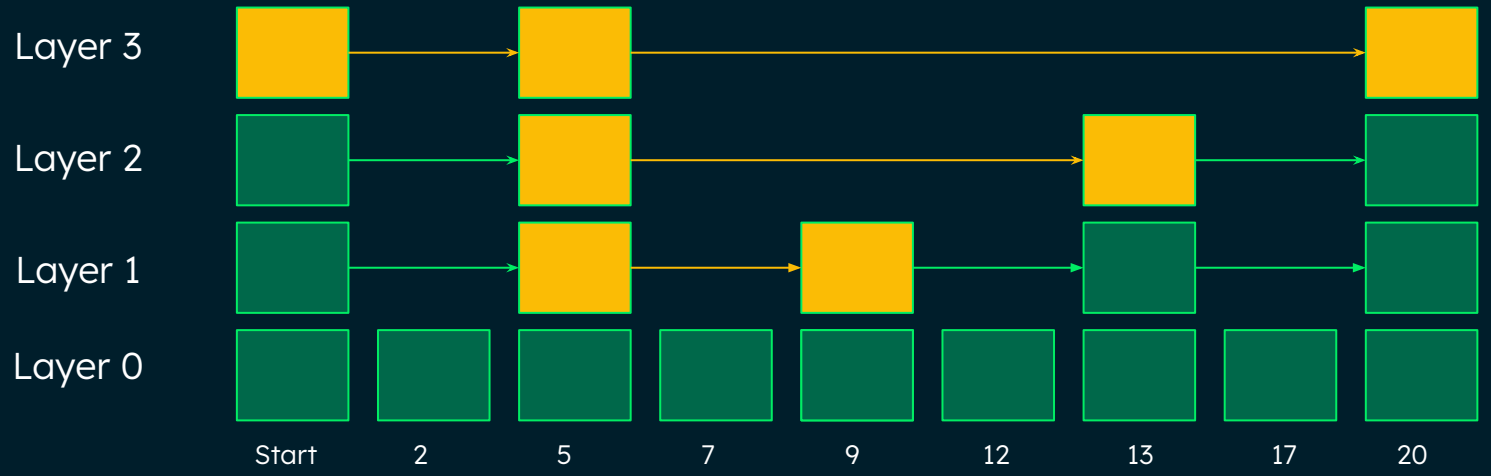
Skip List

Find 9

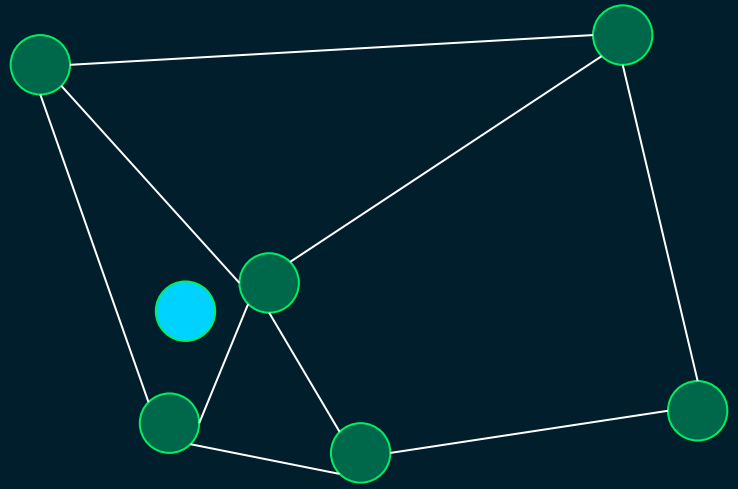


Skip List

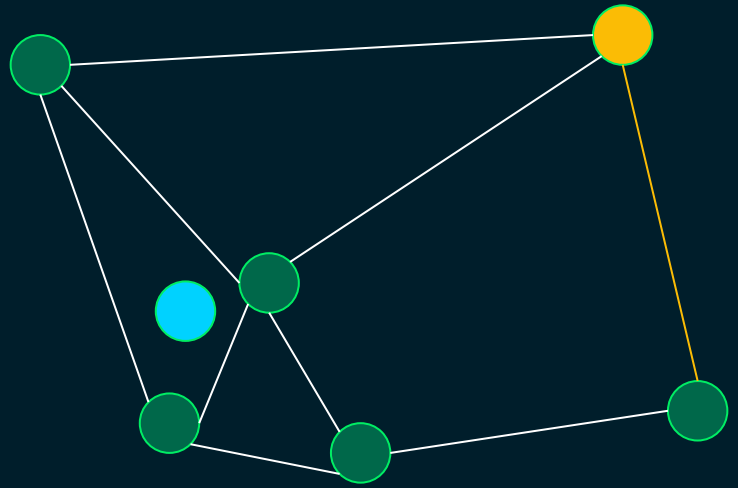
Find 9



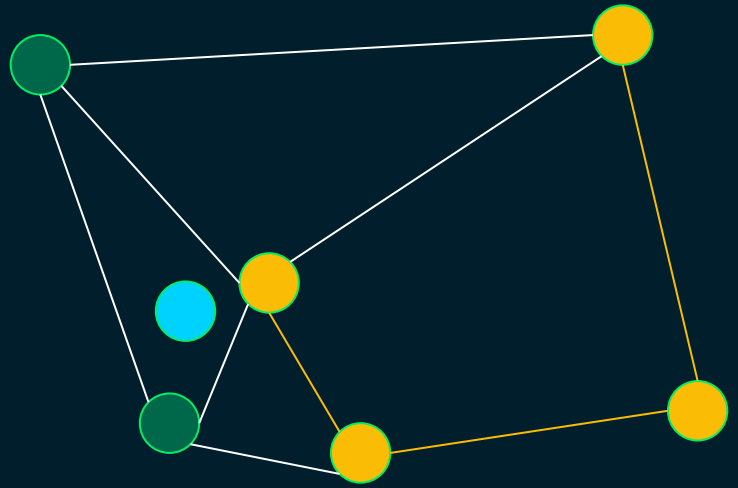
Navigable Small World Graph



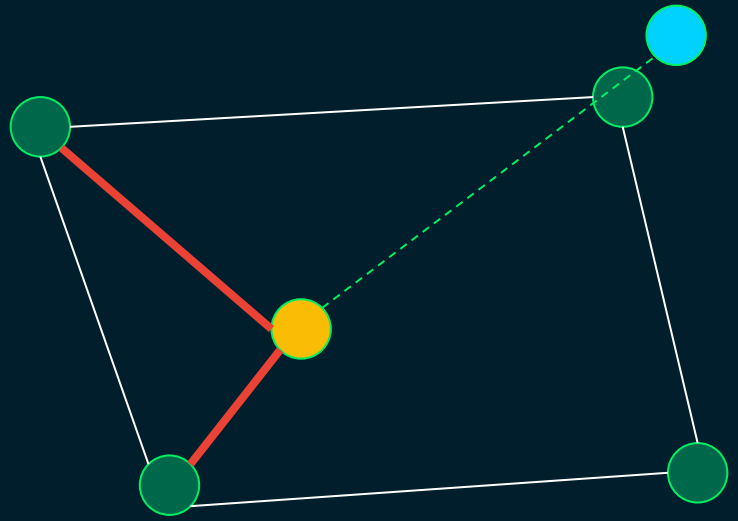
Navigable Small World Graph



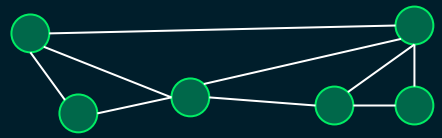
Navigable Small World Graph



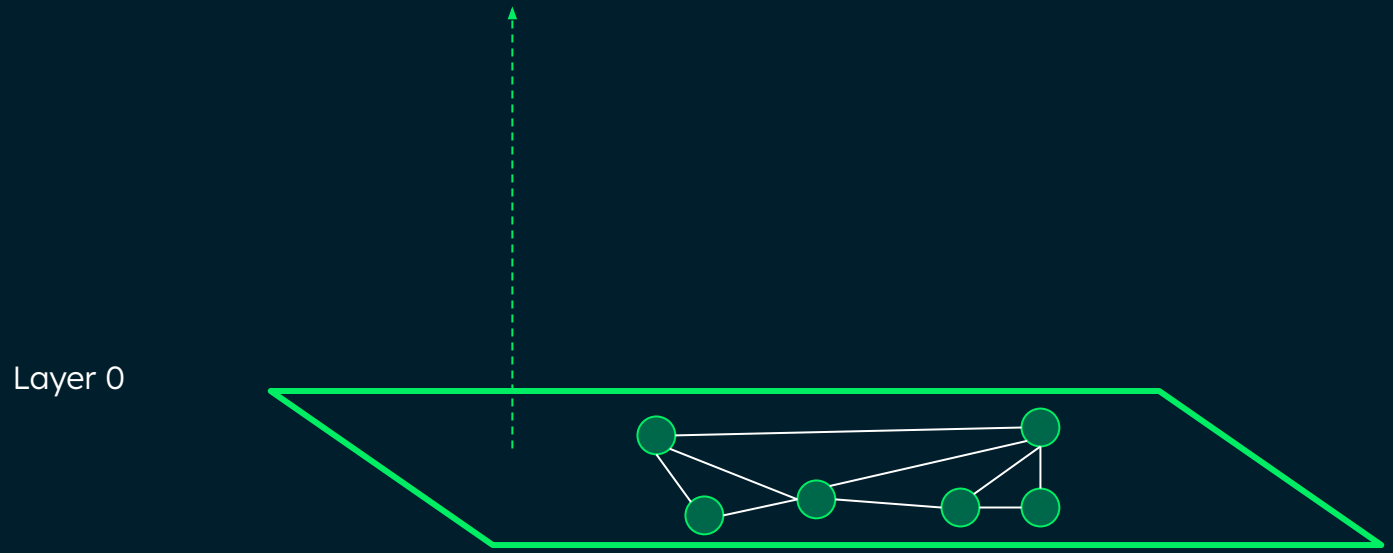
Navigable Small World Graph



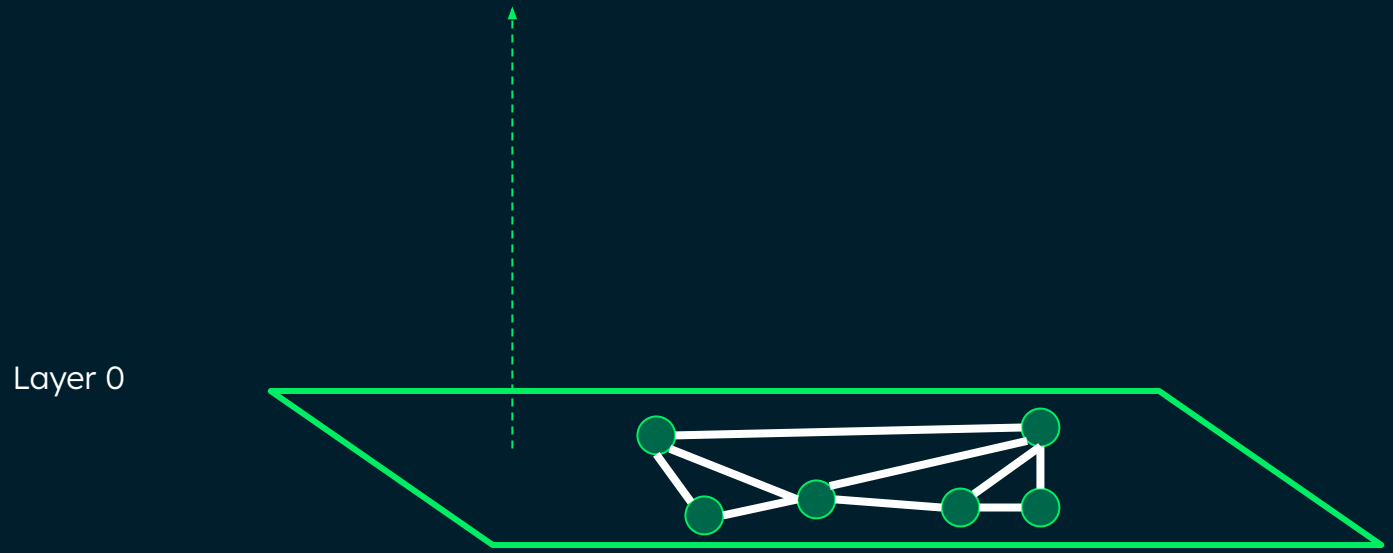
Hierarchical Navigable Small World Graph



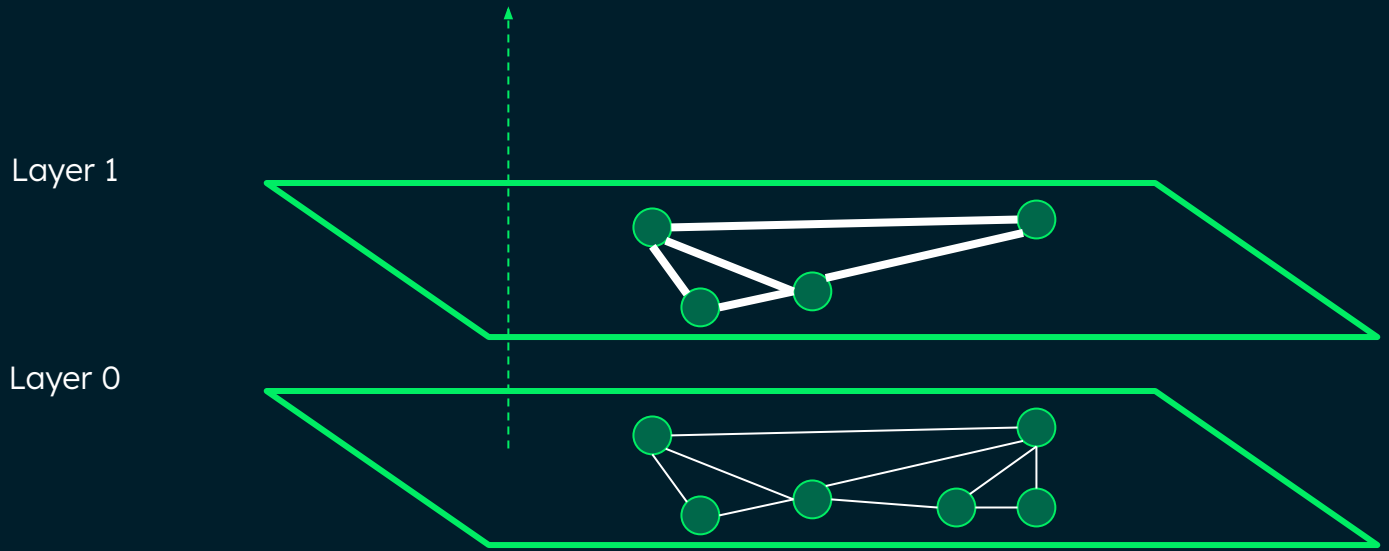
Hierarchical Navigable Small World Graph



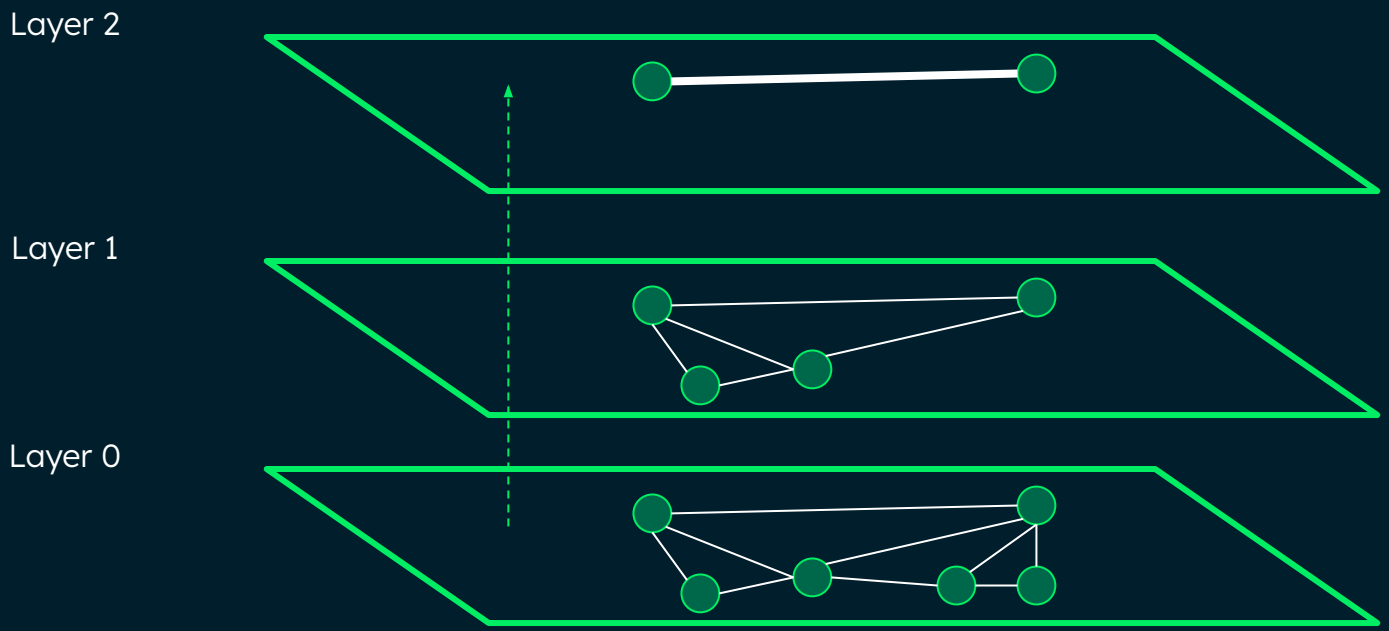
Hierarchical Navigable Small World Graph



Hierarchical Navigable Small World Graph

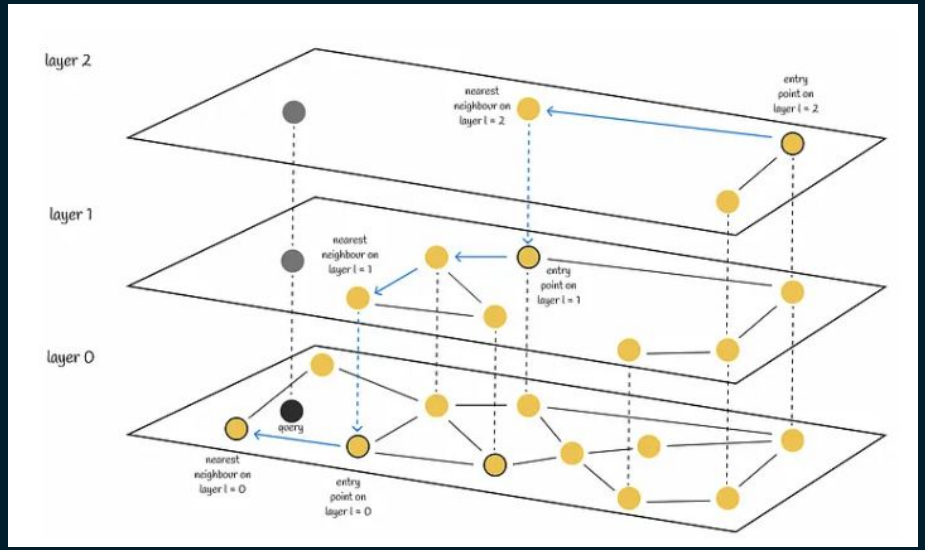


Hierarchical Navigable Small World Graph



How vector search works in MongoDB

SKILL

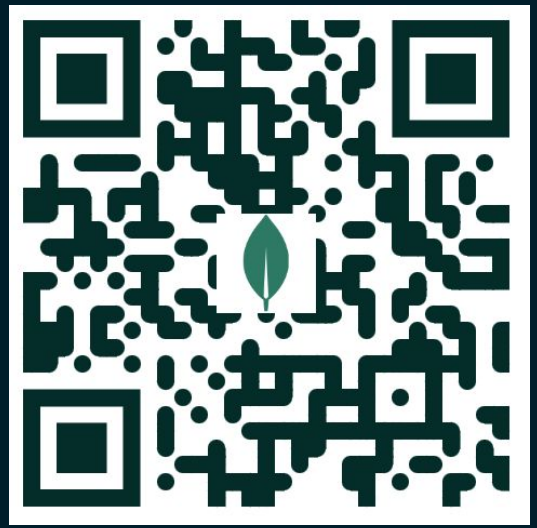


Source: Towards Data Science

Hierarchical Navigable Small Worlds

- Creates layered, connected graphs with vectors as nodes, edges created based on distance in vector space
- Coarse search at top layers, refinement at lower layers
- Efficient due to approximate nearest neighbor search (ANN)

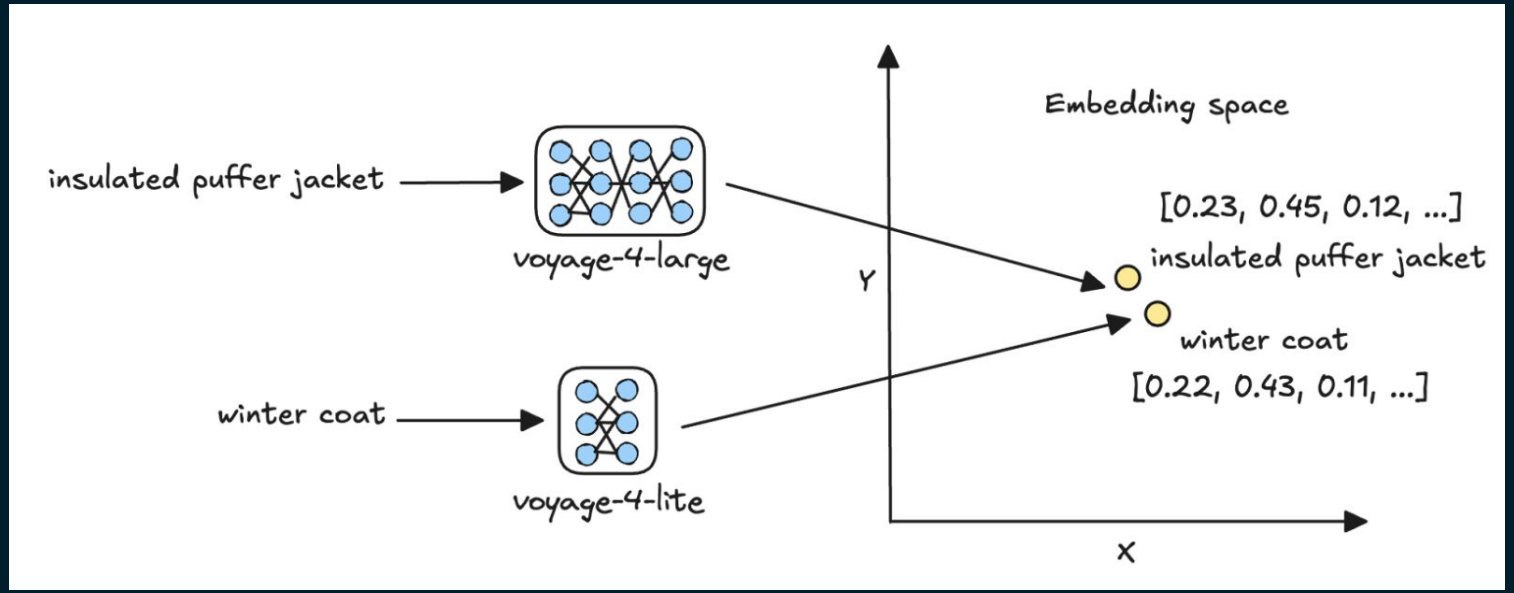
HNSW deep-dive



mdb.link/hnsw-deepdive

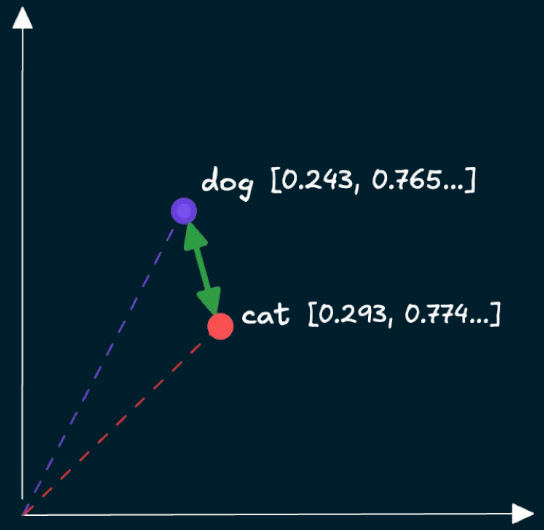
Asymmetric retrieval using Voyage 4

SKILL



Read more: mdb.link/voyage-4

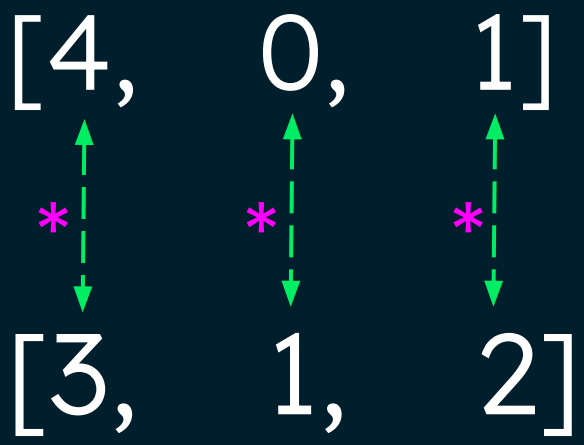
Calculating distance in vector space



Euclidean Distance

Measures absolute distance between vectors

Calculating distance in vector space



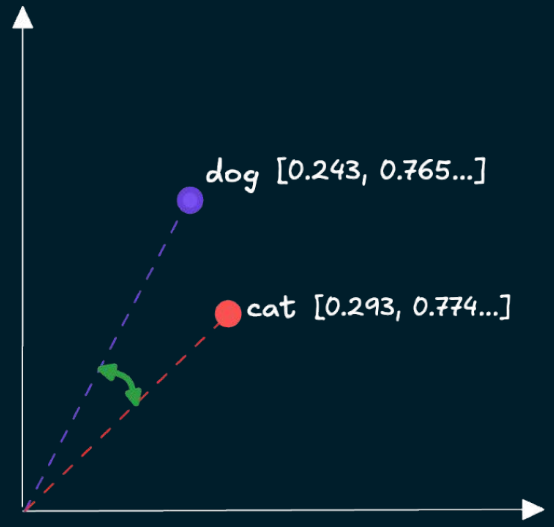
Dot product

Vector multiplication as a measure of alignment

$$12 + 0 + 2 = 14$$

Calculating distance in vector space

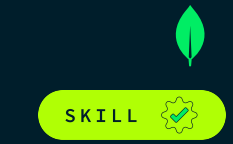
SKILL



Cosine similarity

Measures the angle between vectors

Recap



- Vector search retrieves documents closest to the query embedding in vector space.
- Use compatible embedding models to embed the data and the user queries.
- Distance in vector space is calculated using mathematical functions.
- Cosine similarity works well with most embedding models.

Vector search in MongoDB

1. Generate embeddings 
2. Create a vector search index
3. Send a vector search query

Create a vector search index

ERAS API

```
{
  “fields”:[
    {
      “type”: “vector”,
      “path”: “embedding”,
      “numDimensions”:1024,
      “similarity”: “cosine”
    },
    ...
  ]
}
```

Auto-embedding

```
{
  “fields”:[
    {
      “type”: “autoEmbed”,
      “modality”: “text”,
      “path”: “synopsis”,
      “model”: “voyage-4”
    },
    ...
  ]
}
```

[ANIMATION]

Send a vector search query

SKILL

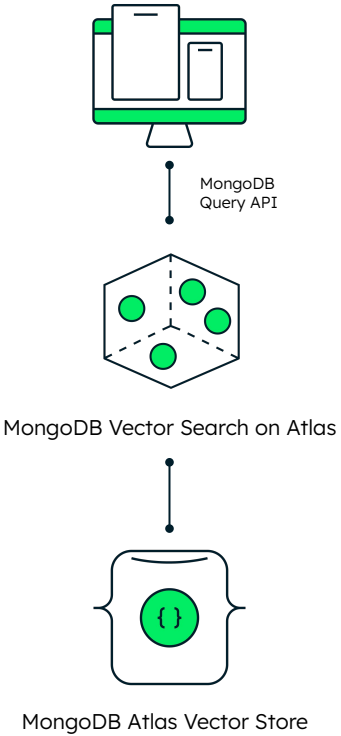
ERAS API

```
[
  {
    “$vectorSearch”: {
      “index”: “vector_index”,
      “path”: “embedding”,
      “queryVector”: [0.02421053, ...],
      “numCandidates”: 200,
      “limit”: 10
    }
  },
  {
    “$project”: {
      “_id”: 0,
      “title”: 1,
      “score”: {“$meta”: “vectorSearchScore”}
    }
  }
]
```

Auto-embedding

```
[
  {
    “$vectorSearch”: {
      “index”: “vector_index”,
      “path”: “synopsis”,
      “query”: {“text”: “A boy in a…”},
      “model”: “voyage-4”,
      “numCandidates”: 200,
      “limit”: 10
    }
  },
  {
    “$project”: {
      “_id”: 0,
      “title”: 1,
      “score”: {“$meta”: “vectorSearchScore”}
    }
  }
]
```

MongoDB Vector Search on Atlas

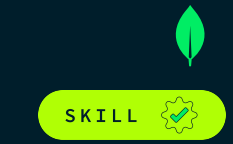


Integrated platform that simplifies your application architecture

- Data is **automatically synchronized** between the database and vector index
- Developers work with database and vector search via the **unified MongoDB Query API**
- **Fully managed** for you so you can focus on your application
- Search nodes **scale your search workloads** independent of the operational database

Vector search simplified — Avoid the tax synchronization — Remove operational heavy lifting

Recap



- To perform vector search in MongoDB, you need to generate embeddings, create a vector search index and send a query.
- The number of dimensions in the embedding vector depends on the embedding model used.
- Vector search should always be the first stage in a vector search aggregation pipeline.

In this exercise, you will:

- Create a vector search index.
- Perform vector search queries.

TODO:

- Navigate to mdb.link/devrel-vector-search
- Complete **Step 5-6** in the notebook

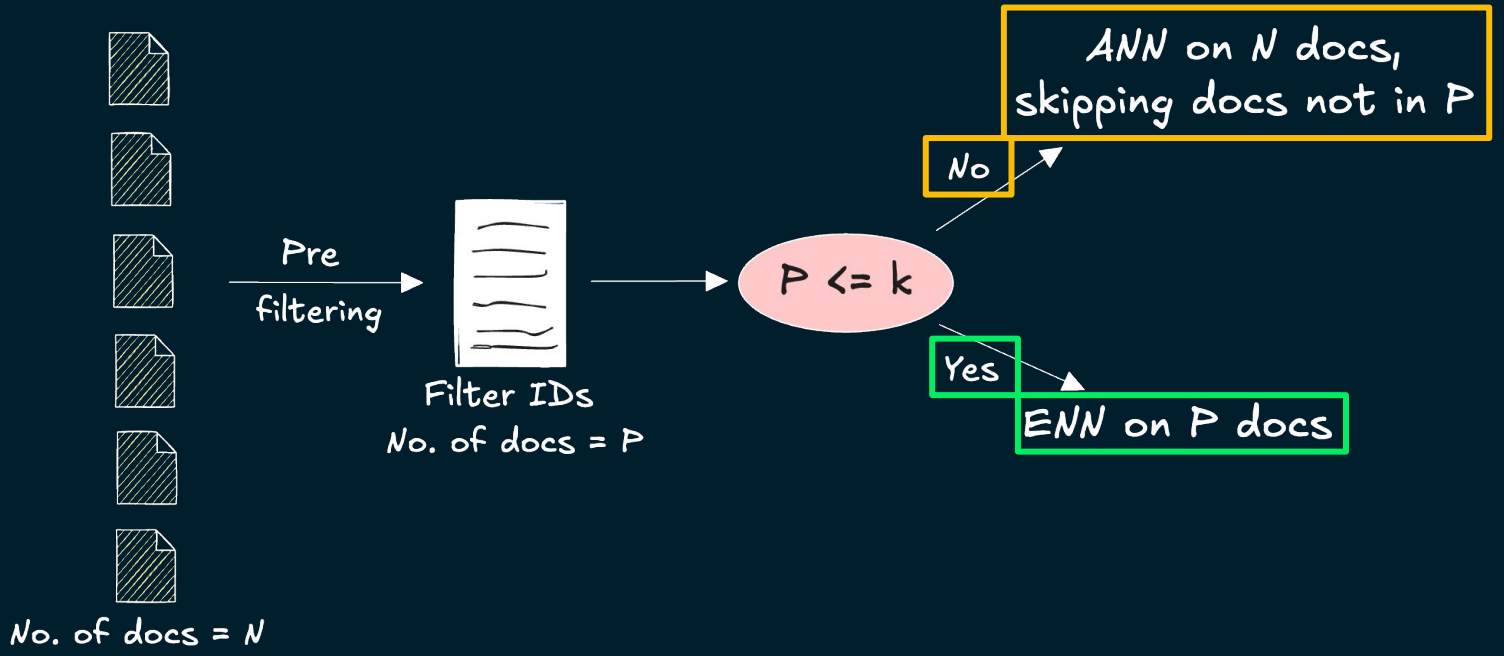
Optimizing Your Vector Search Queries

Pre-filtering

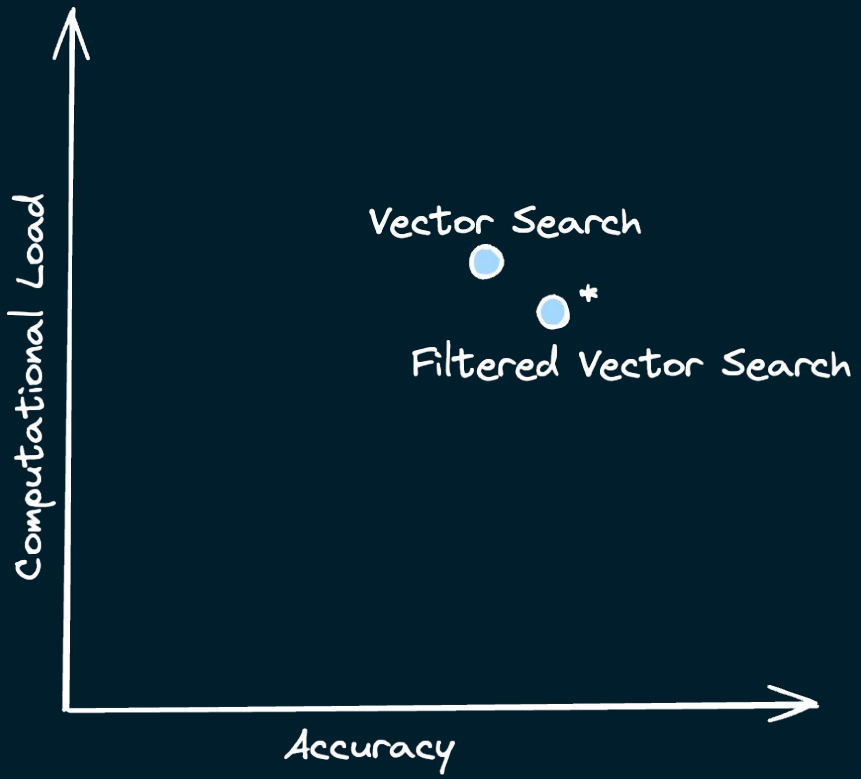
Pre-filtering

SKILL

k = No. of results to return

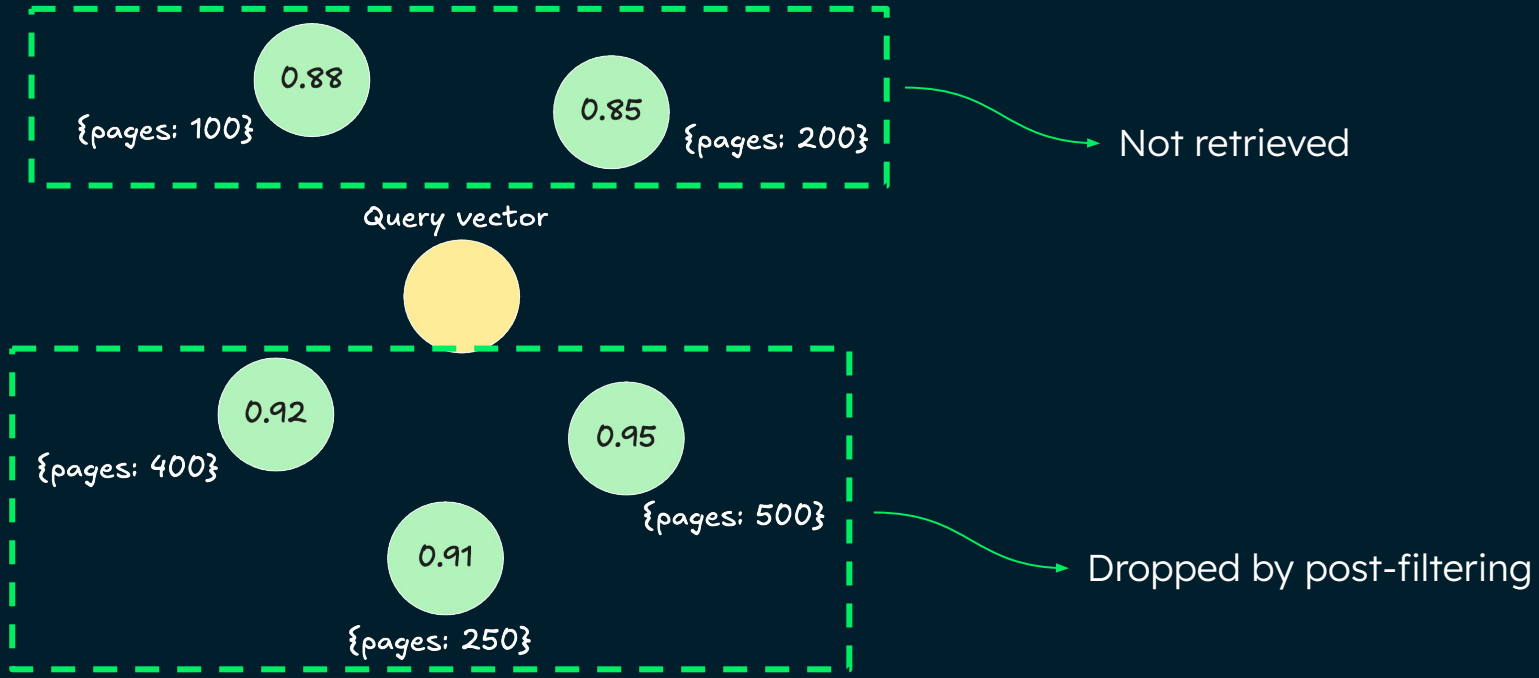


Pre-filtering



Pre-filtering >>> post-filtering

filter : pages <= 200 **k** : 3



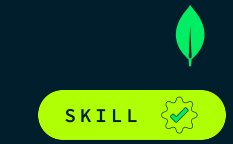
Adding pre-filters to vector search

```
{
  “fields”:[
    {
      “type”: “vector”,
      “path”: “embedding”,
      “numDimensions”: 1024,
      “similarity”: “cosine”
    },
    {
      “type”: “filter”,
      “path”: “pages”
    },
    ...
  ]
}
```

Adding pre-filters to vector search

```
pipeline = [
        {
          “$vectorSearch”: {
            “index”: “vector_index”,
            “path”: “embedding”,
            “filter”: {“pages”: { “$lte”: 200 } },
            “queryVector”: [0.02421053, -0.022372592,...],
            “numCandidates”: 20,
            “limit”: 10
          }
        },
        {
          “$project”: {
            “_id”: 0,
            “Content”: 1,
            “score”: {“$meta”: “vectorSearchScore”}
          }
        }
    ]
```

Recap



- Use pre-filters to filter vector search results based on certain criteria.
- Pre-filtering can slow down the vector search if the filters are too restrictive.
- Post-filtering is generally not recommended.

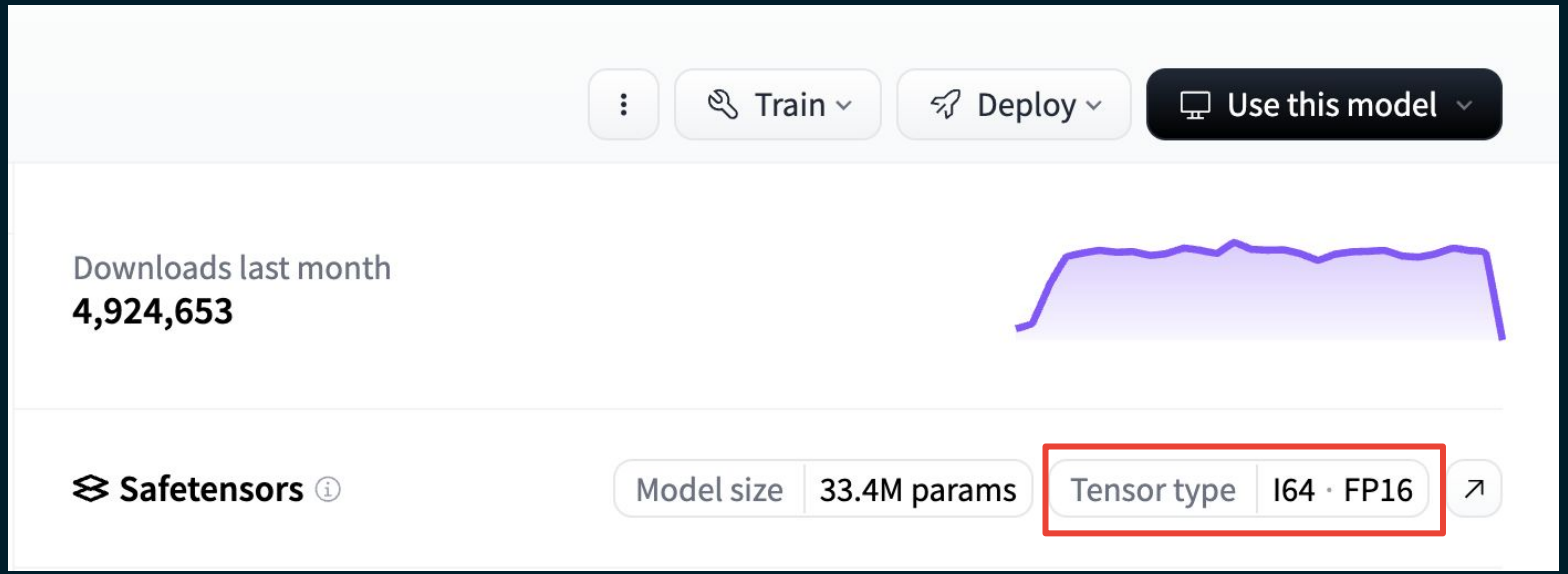
In this exercise, you will:

- Combine pre-filtering with vector search.

TODO:

- Navigate to mdb.link/devrel-vector-search
- Complete **Step 7** in the notebook

Vector Quantization


Train
Deploy
Use this model
Downloads last month
4,924,653
Safetensors
Model size
33.4M params
Tensor type
I64 · FP16

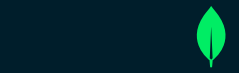
Data type

FP/float: Floating point

int: Integer

No. of bits required to store each number

32 bits = 4 bytes



20 * 1024 * 4 ≅ 81920 bytes ≅ 82 KB

What if you had 100 million embeddings?

100000000 * 1024 * 4 ≅ 410 GB

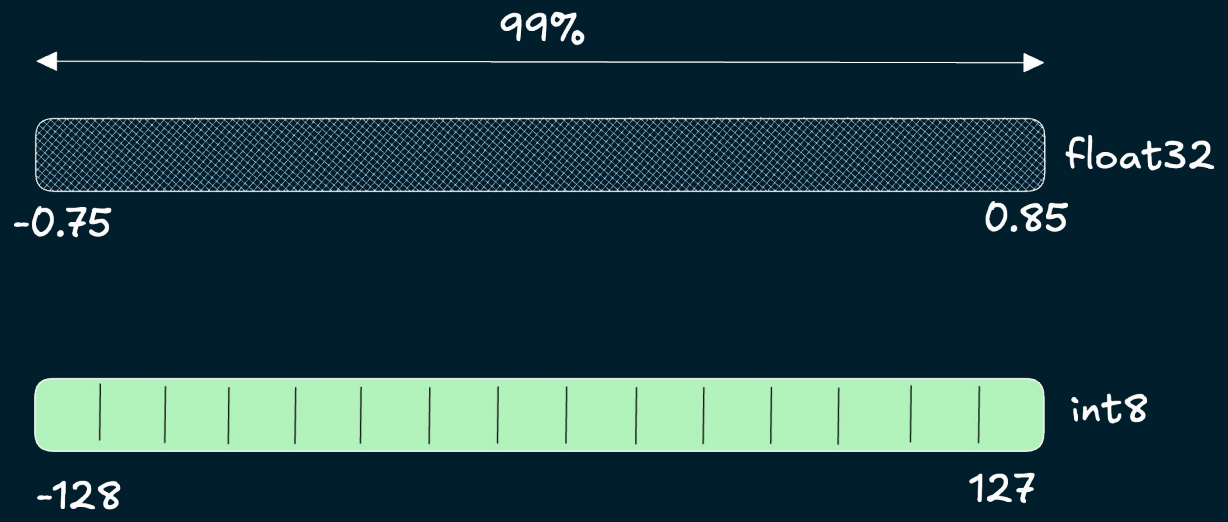
Vector quantization

Quantization is the process of shrinking full-fidelity vectors into fewer bits

- **Scalar**
- **Binary**

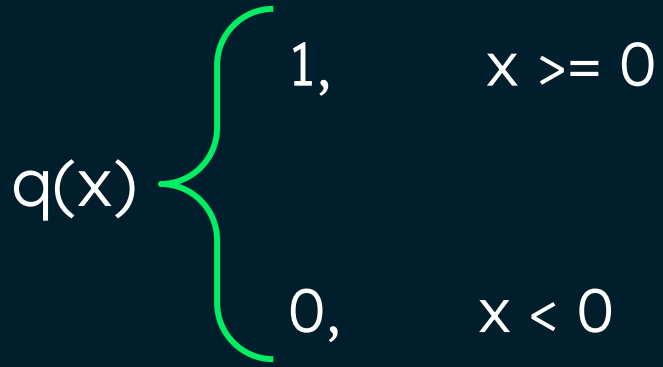
Scalar quantization

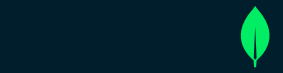
Takes each vector dimension and buckets it into a smaller set of discrete integers



Binary quantization

Sets each vector dimension to a binary value based on a threshold





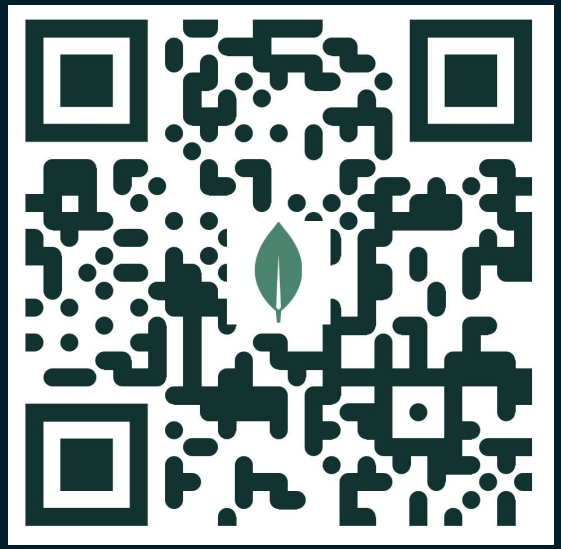
Data size: 3M

Test	Memory and index savings	% of default performance
Full-fidelity	1x	100%
Scalar quantization	3.75x	97%
Binary quantization	24x	95%

Auto-quantization in MongoDB

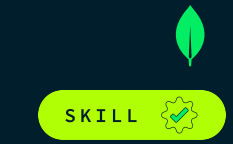
```
{
  “fields”:[
    {
      “type”: “vector”,
      “path”: “embedding”,
      “numDimensions”:1024,
      “similarity”: “cosine”,
      “quantization”: “scalar”
    },
    ...
  ]
}
```

Read more about quantization



mdb.link/quantization

Recap



- Vector quantization is the process of shrinking full-fidelity vectors into fewer bits.
- Quantization reduces retrieval latency and storage costs for a small dip in accuracy.
- Scalar quantization buckets each vector dimension into a smaller set of discrete integers.
- Binary quantization sets each vector dimension to a 0 or 1.

In this exercise, you will:

- Enable auto-quantization in MongoDB.

TODO:

- Navigate to mdb.link/devrel-vector-search
- Complete **Step 8** in the notebook

SKILL CHECK

Vector Search Fundamentals

- Understand Semantic Search
- Store Embeddings for Your Data
- Perform a Vector Search

https://mdb.link/devday-chicago-26-vector-search

Questions?

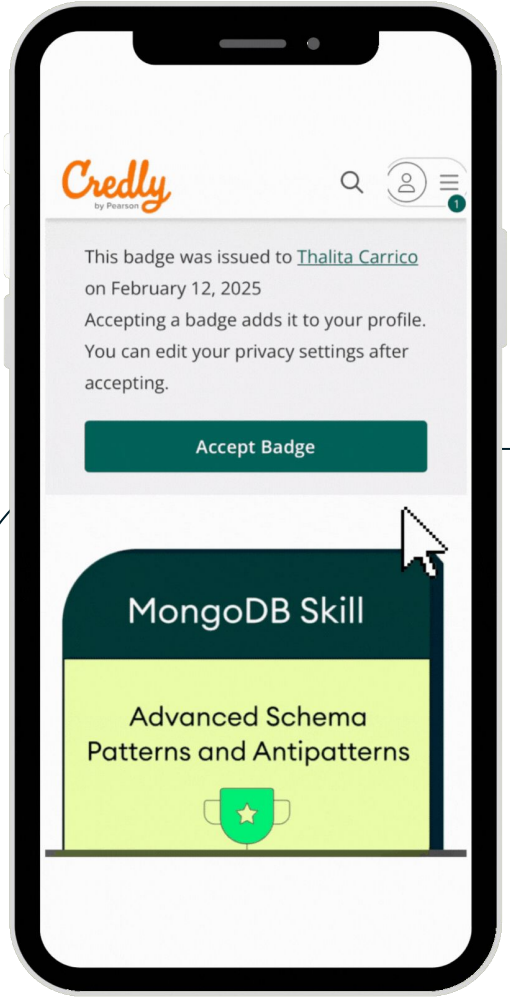
Ask your neighbor

Raise your hand

Let’s do it together!

Share Your Badge

Use **#MDBDAYCHICAGO**



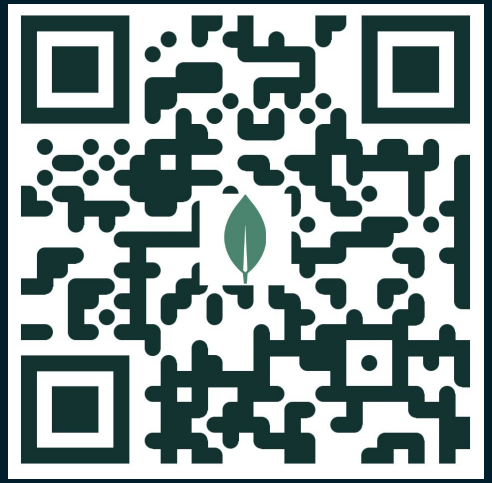
Keep Learning

Every badge you earn elevates your skillset, empowering you and your team to accelerate and optimize application development and operations.

Explore topics like:

- Data modeling
- Generative AI
- Aggregation
- Querying
- Security
- Indexing
- Sharding
- Monitoring, Tuning, and Automation
- Search
- Performance at Scale

Check out our GenAI Examples repo!



mdb.link/genai-examples

Check out our AI Learning Hub!

mdb.link/ai-learning-hub

Q & A

Thank You!