Who Am I?

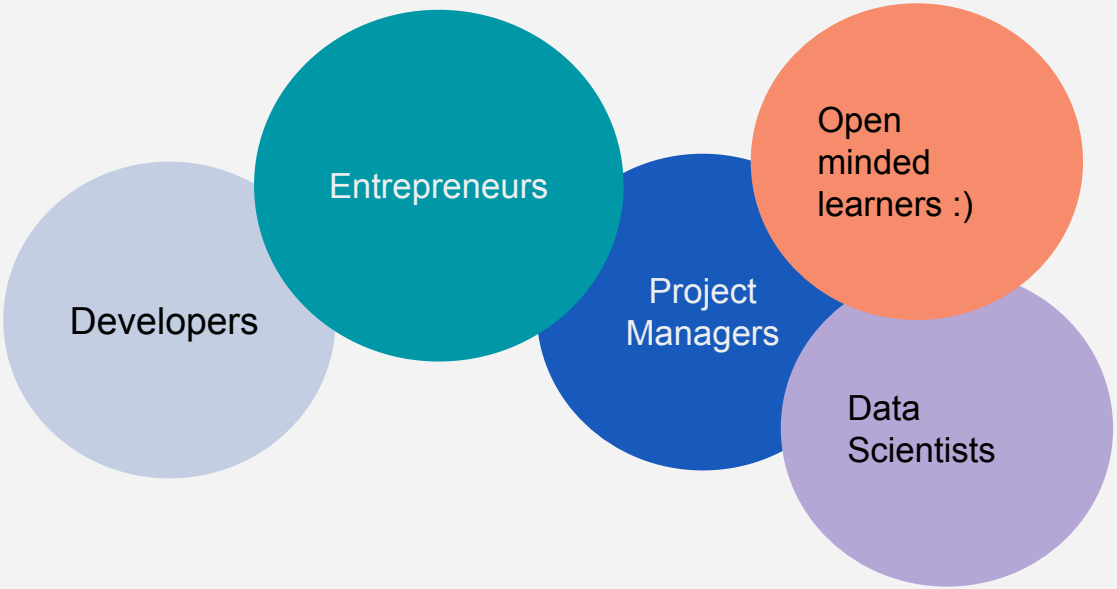
Paulo Dichone

Software, Cloud, AI Engineer
and Instructor

What Is This Course About?

- Vector Databases - Fundamentals (Deep Dive)

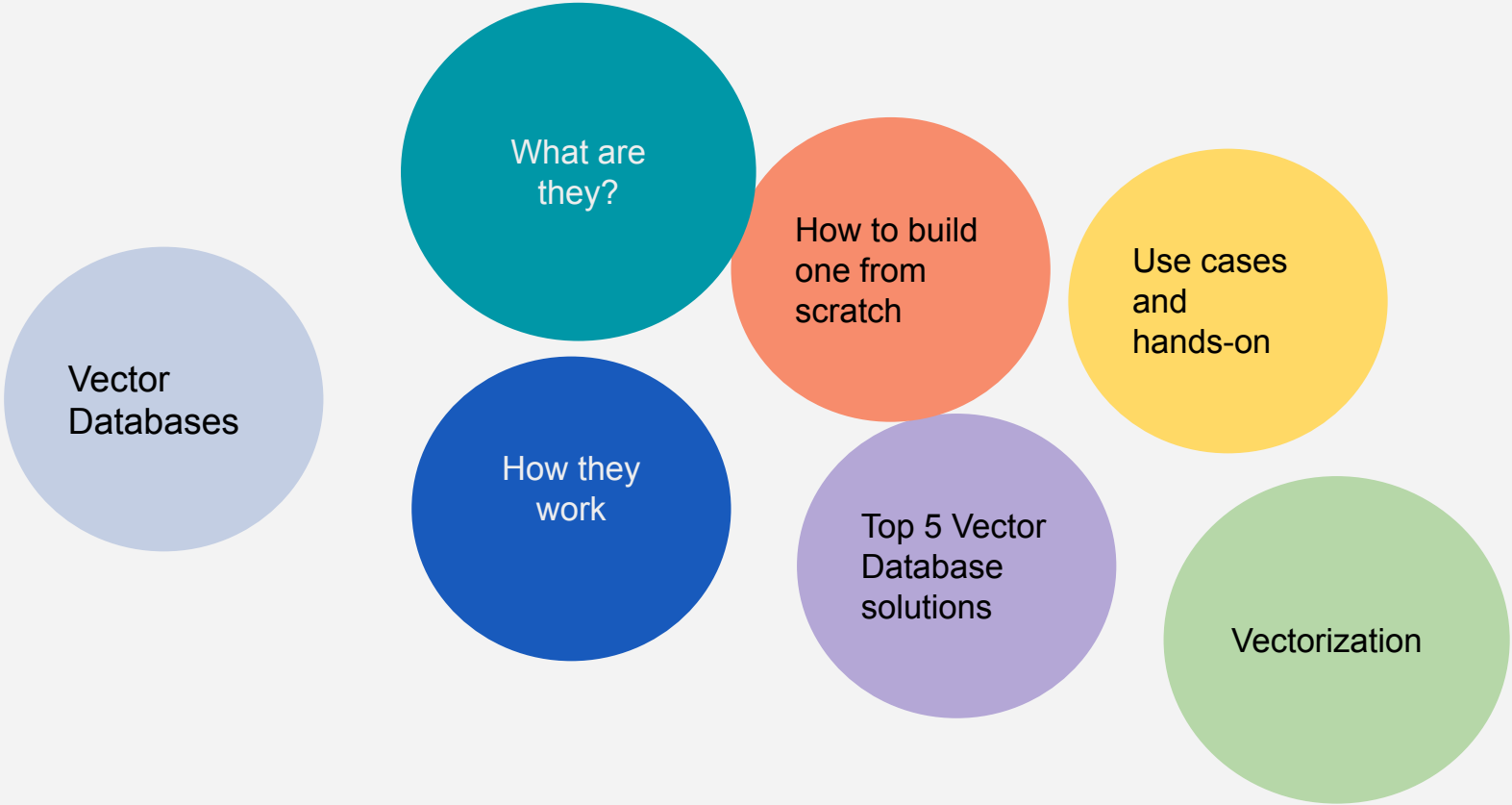
Who Is This Course For



Course Prerequisites

1. Know Programming (highly *preferred…)*
    a. *There will be some Python code*
2. *This is not a programming course*
3. Willingness to learn :)

What you'll learn



Course Structure

Development Environment setup

- Python
- VS Code (or any other code editor)
- OpenAI API Account and API Key

Set up OpenAI API Account

OpenAI API - Dev Environment Setup

Python (Win, Mac, Linux)

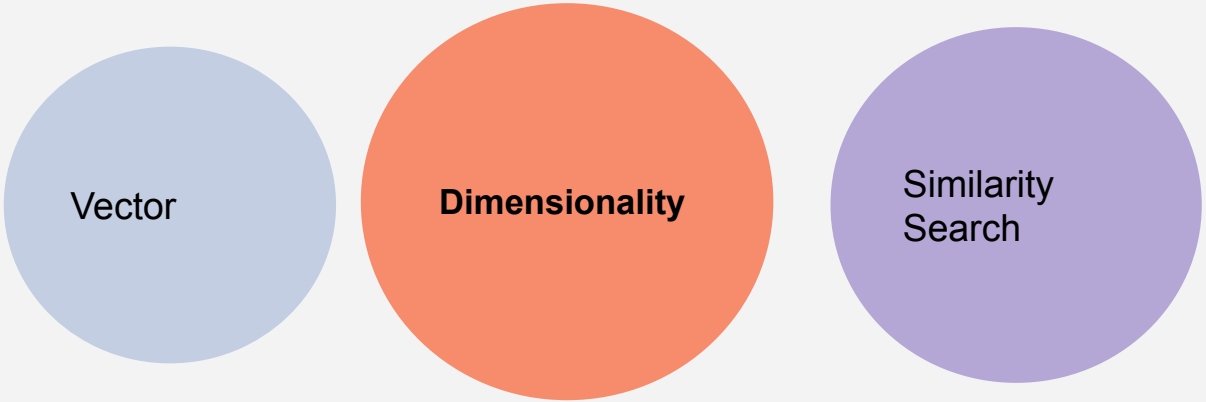
https://kinsta.com/knowledgebase/install-python/

Introduction To Vector Databases

- What is a vector database?
- Why vector databases?
- Limitation of traditional databases

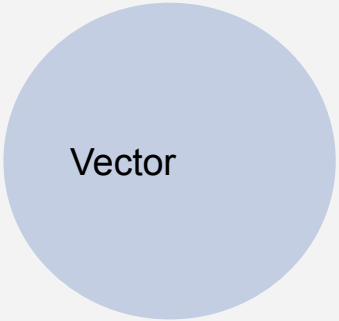
What is a Vector Database?

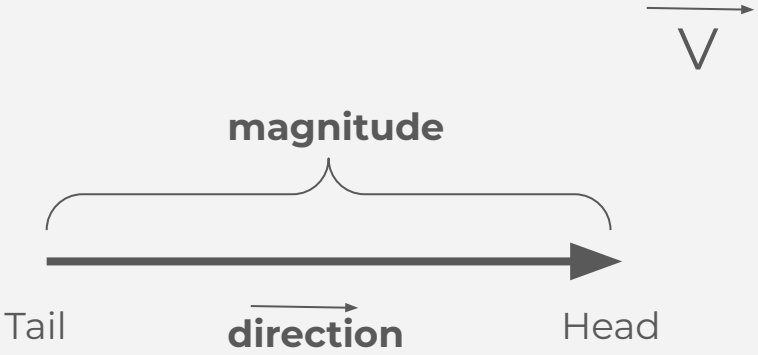
A vector database encodes information as *vectors* in a multi-dimensional space to perform high-efficient queries based on similarity.



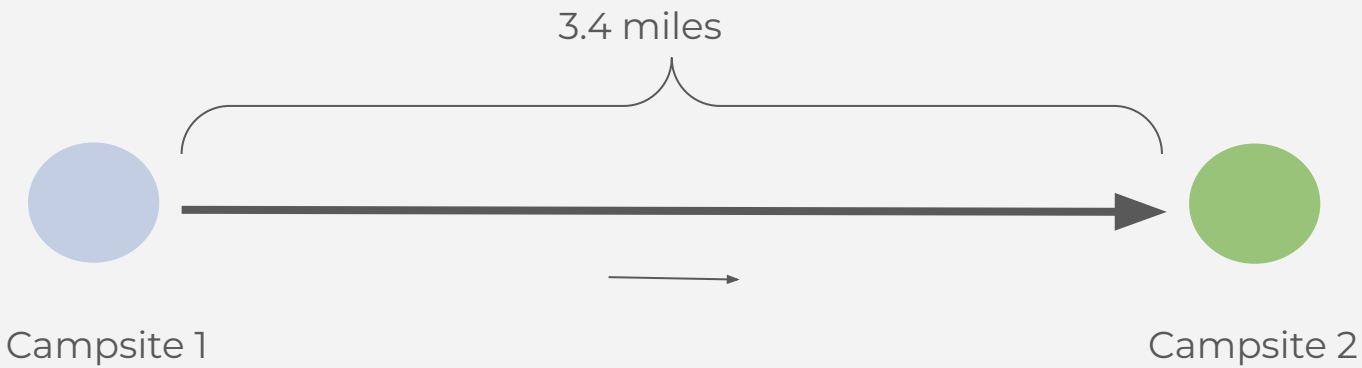
What is a Vector Database?

A *vector* - what is it?





Vectors



Direction - *toward camp 2*
Magnitude - *3.4 miles*

Why Vector Databases?

How data “shows up” in the world?

80% or more of data is unstructured

Vector databases are specifically excellent for working with these types of data (*actually, the only type of databases that can work with unstructured data)*

Why?

Why Are Vectors Used in a Vector Database?

Because...

Vector databases turn....

Into

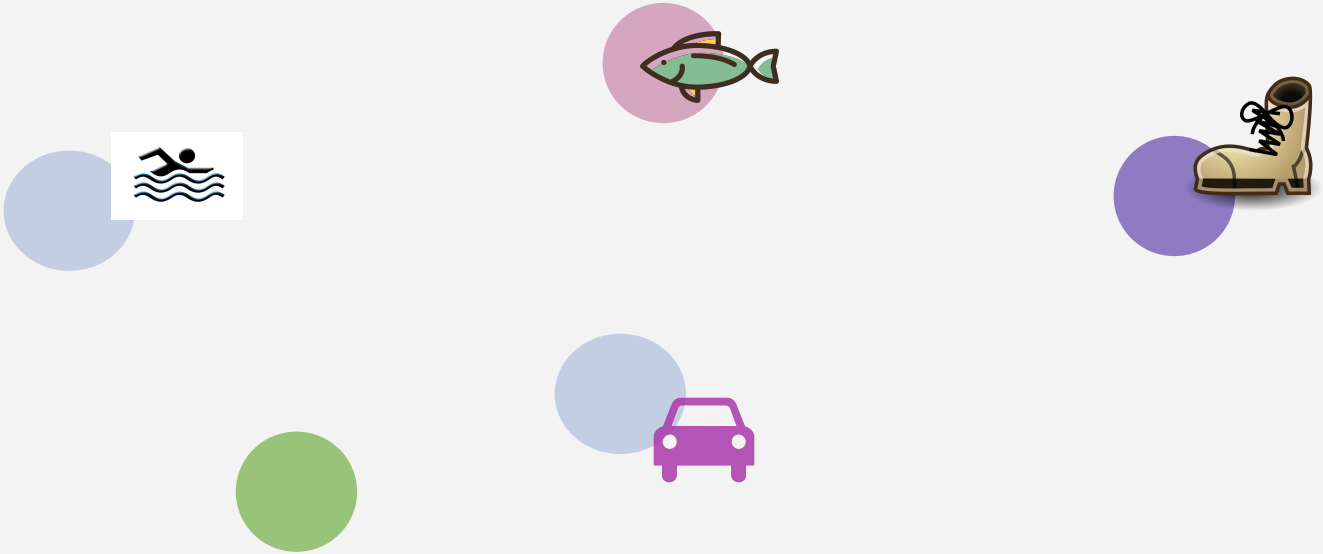
Numbers (vectors)

[**-0.05**, -0.0955,..., 0.0722]

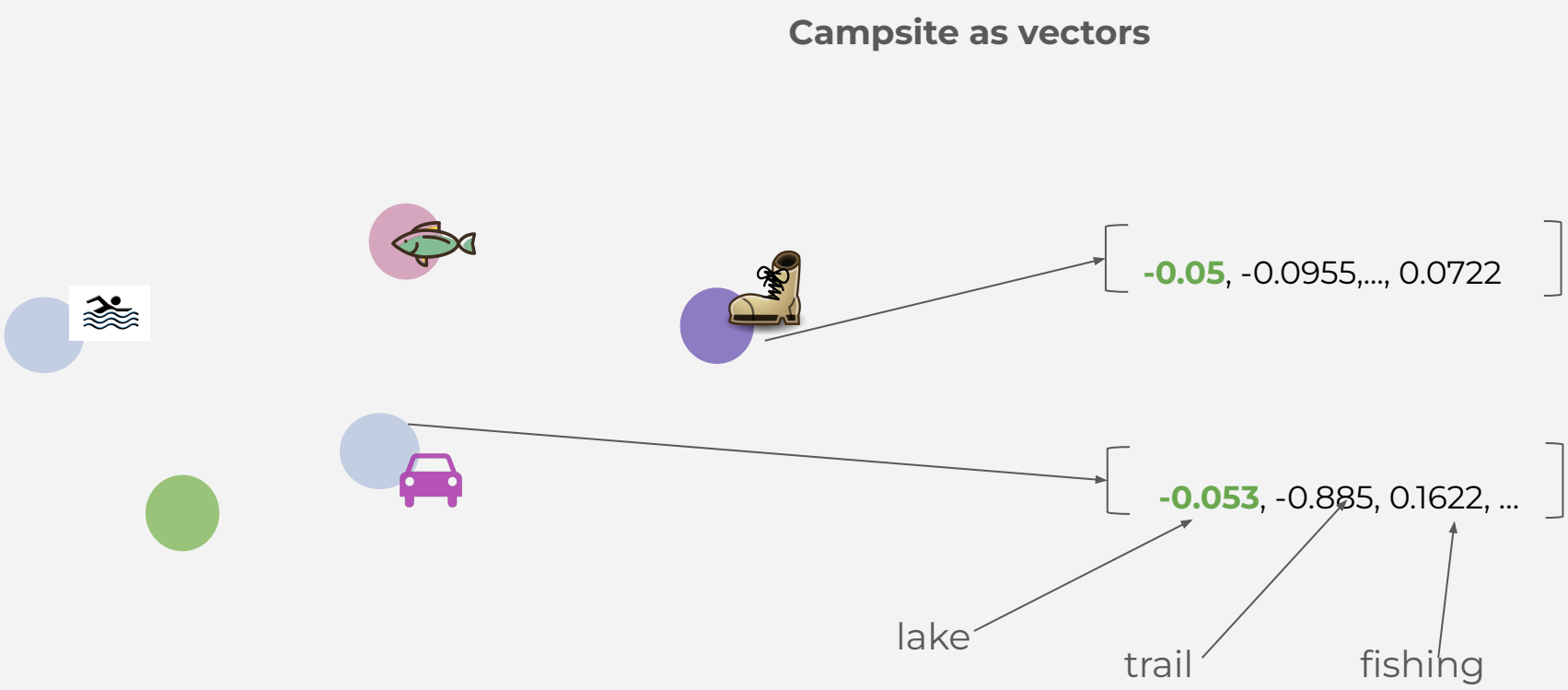
[**-0.053**, -0.885, 0.1622, ...]

Why Are Vectors Used in a Vector Database?

We are back to the first question... and the campground



Vectors in Action



Benefits - Quick Search for the Best Campsite

lake

[**-0.05**, -0.0955,..., 0.0722]

water

[**-0.053**, -0.885, 0.1622, ...]

Rapid discoverability
Efficient organization

Why are Vectors Used in a Vector Database?

1. **Efficient Representation of Complex Data**
    a. Dimensionality - representing data in high-dimensional space
    b. Uniformity - data can be converted into a uniform format (numerical vectors)
2. **Enabling Similarity Search**
3. **Leveraging Machine Learning Models**
4. **Optimizing Performance and Scalability**
5. **Improving User Experience**
    a. Real-time interaction (recommendations, search results or data analysis outputs)

lake

[**-0.05**, -0.0955,..., 0.0722]

water

[**-0.053**, -0.885, 0.1622, ...]

Traditional Databases

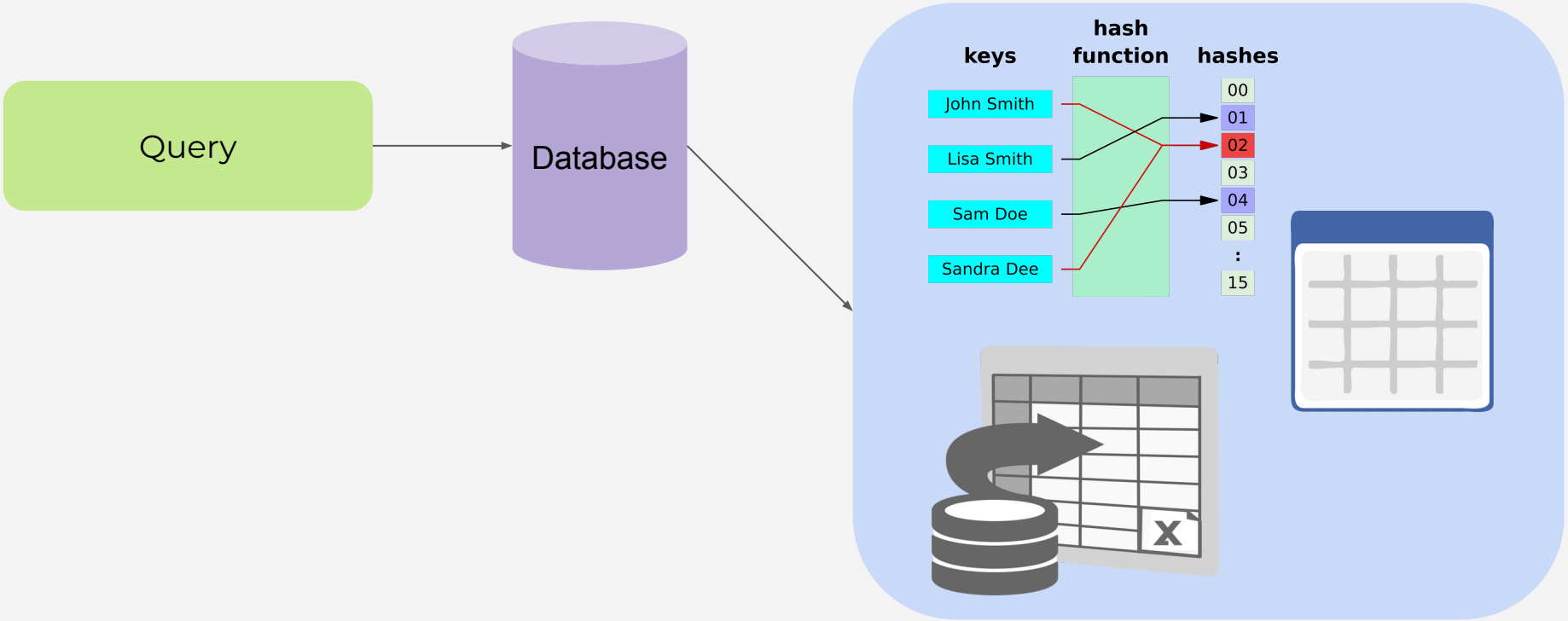
In contrast to Vector Databases - RDBMS

Key characteristics

- **Structured** data: predefined columns and rows
- **Schema-based**: database structure must be defined before hand.
- **Data manipulation and querying:** manipulation through SQL
- **ACID Compliant:** Atomic, Consistency, Isolation, Durability
- **Indexing:** to speed up data retrieval

Traditional Databases

How data search works in traditional databases:

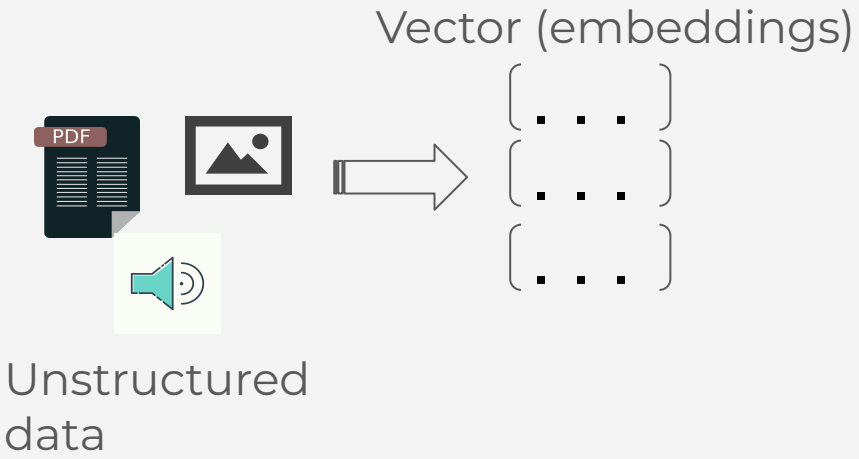


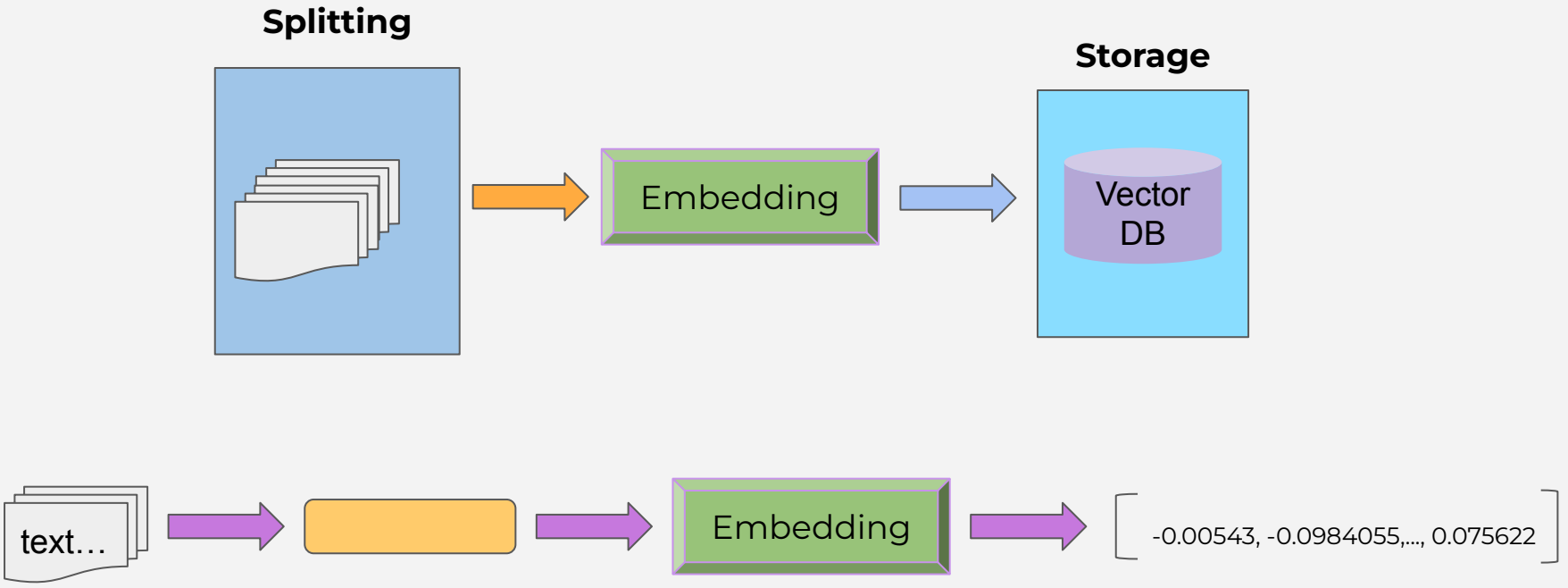
Traditional Databases

Limitations

- **Scalability**: hard to deal with complex queries across large tables
- **Flexibility**: changing DB's schema can be disruptive
- **Handling Unstructured Data:** not well-suited for handling unstructured data (images, text, audio, video)

Transforming Unstructured Data into Vectors - Deep dive





Embedding vector keeps the content and the meaning of the text
Text with similar content and meaning will have similar vectors

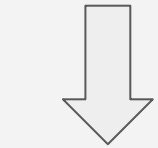
Text with similar content and meaning will have similar vectors

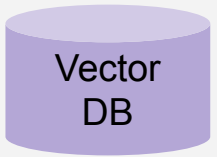
Cat

[**-0.05**, -0.0955,..., 0.0722]

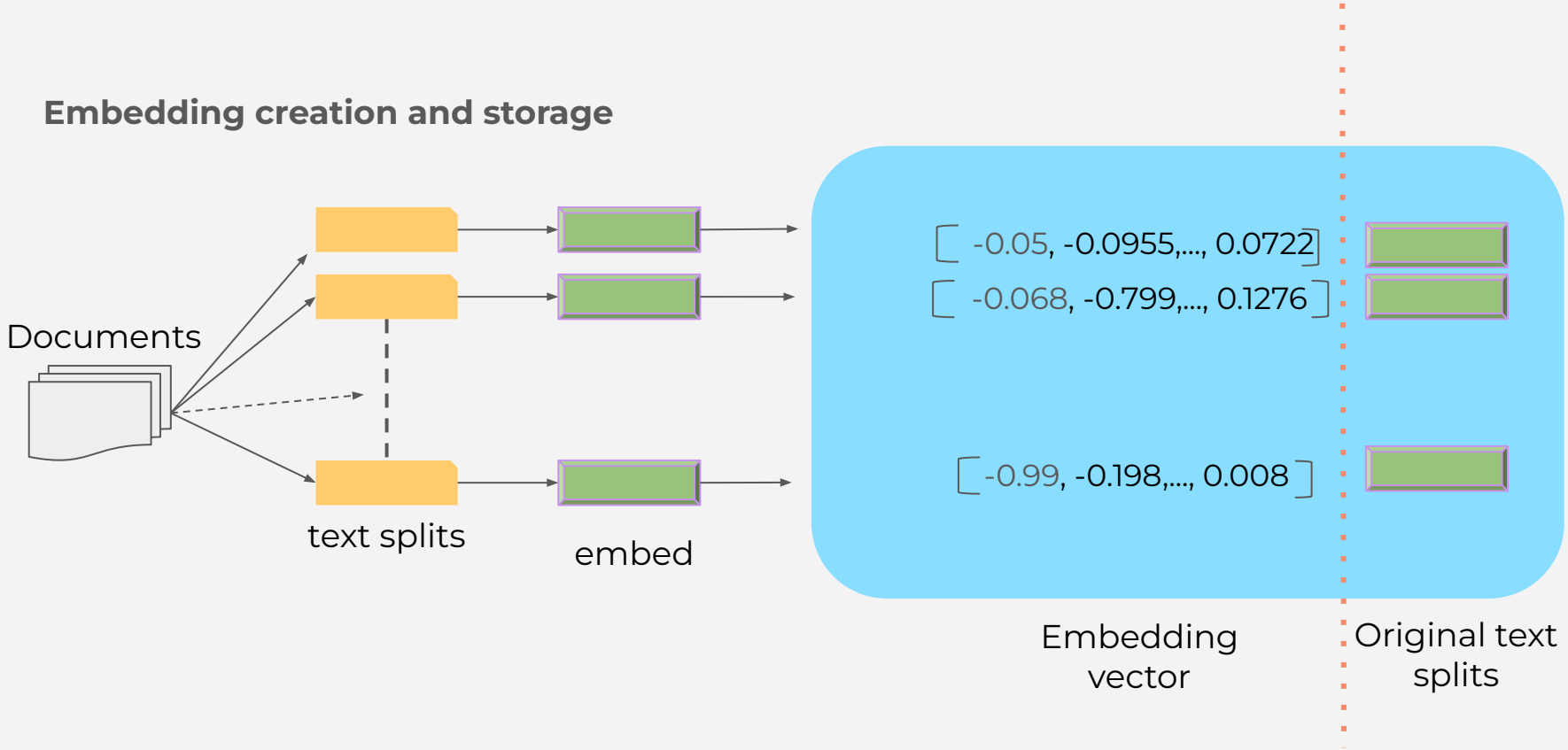
kitty

[**-0.053**, -0.885, 0.1622, ...]

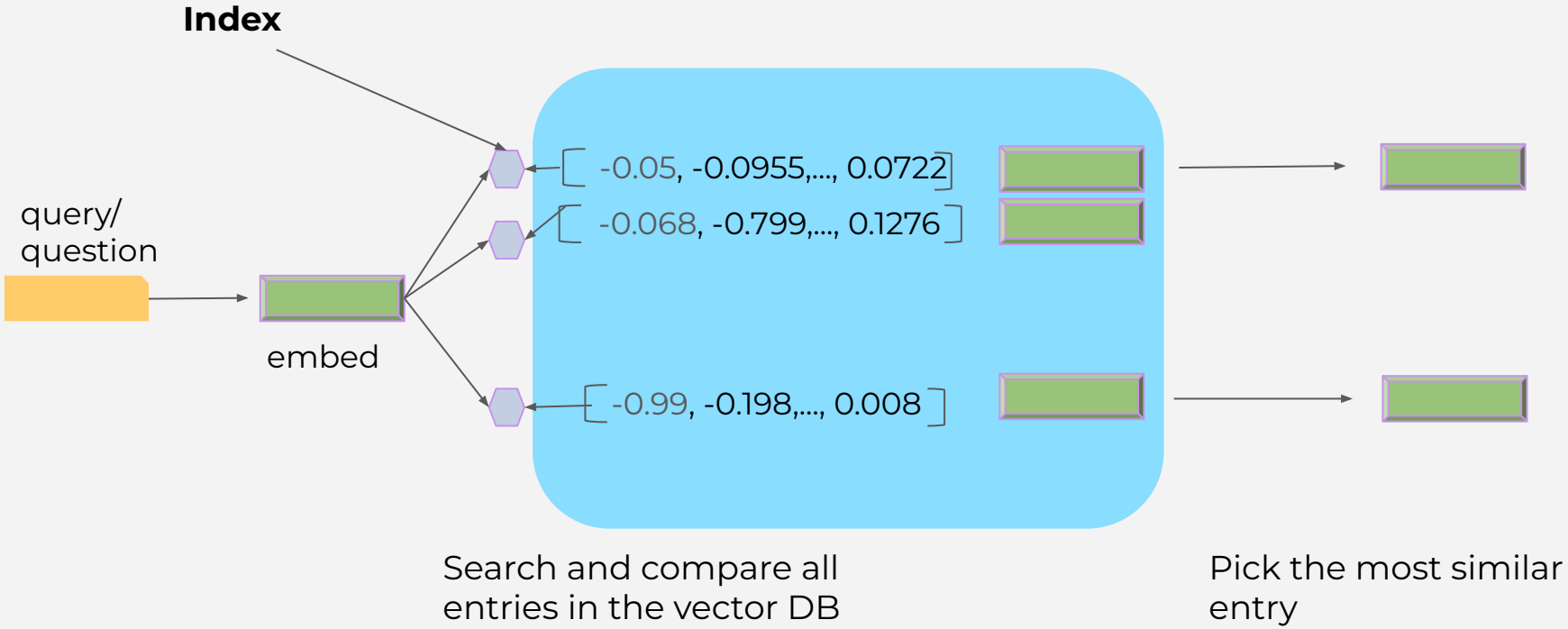




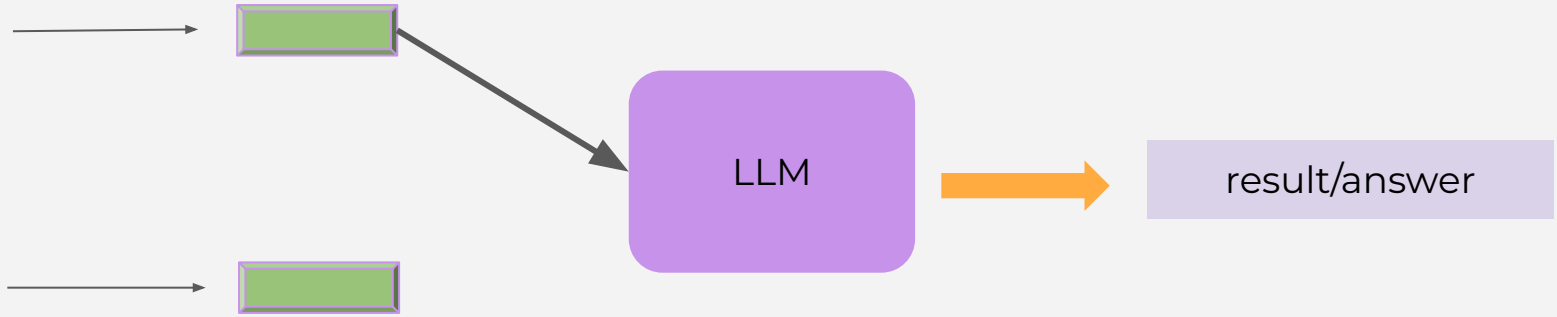
Vector database (vectorstore) - full overview



Vector Database - querying the vector database



Vector Store - Processing with LLM



Take the most similar entry found and pass it on to the LLM along with the question/query...

Pick the most similar entry

Embeddings vs vectors

They essentially refer to the same thing... BUT they have distinct definitions and roles.

$\vec{v}$ Mathematical representation of data in an n-dimensional space (each dimension == a feature of the data)

Embeddings A specific type of vector used in Machine Learning and Artificial Intelligence

Embeddings vs vectors

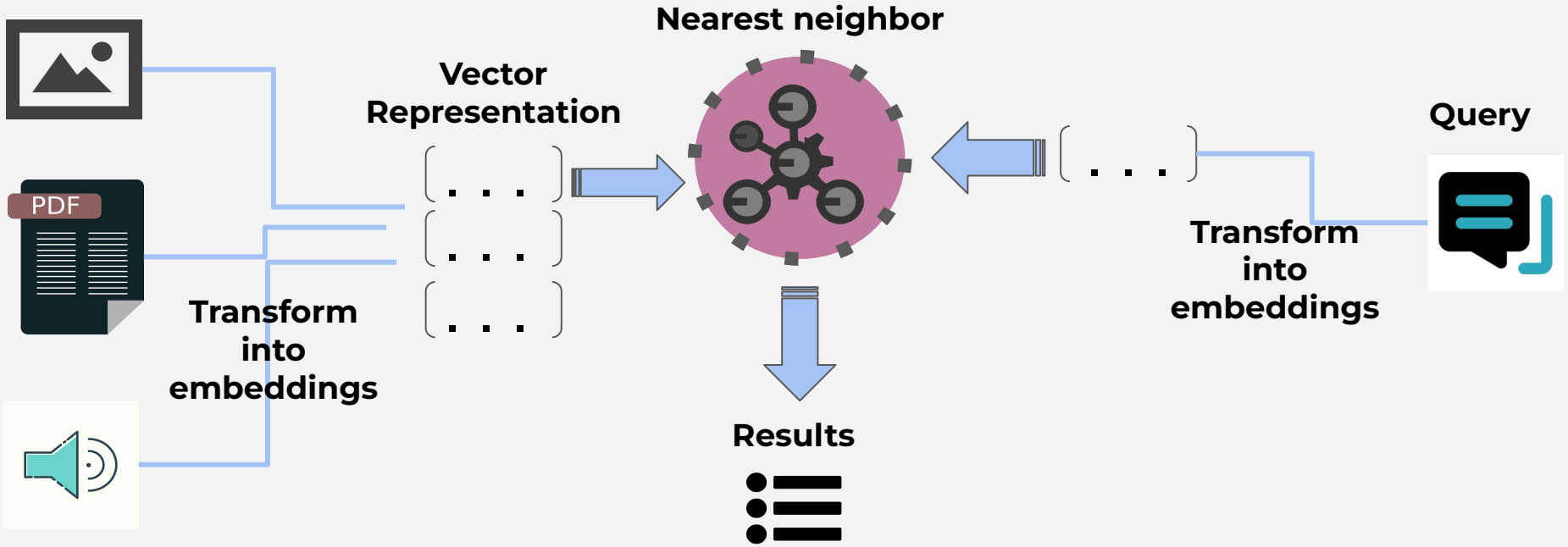
Key differences and relation

Vectors are generic and used for a wide array of applications (handle numerical computations...)

Embeddings map raw data into a vector space (**preserves** semantic relationships - **meaning**)

Bottomline: embeddings are **vectors,** but not all vectors are embeddings. Embeddings are vectors that encodes semantic similarities between the items they represent (great for AI applications)

Vector databases - how they work



Vector Databases

Advantages

- **Data Representation as Vectors**: vector is vectorized which brings lots of benefits for searching
- **Similarity Search**: finding data points closest to a given query vector.
- **Efficiency in High-Dimensional Searches:** use of specialized indexing structures that are highly optimized
- **Handling Unstructured Data:** vector database are made to deal with unstructured data!
- **Schema-less Design:** don’t require schema - allowing more flexibility in handling various data types and structures

Vector Databases Use cases

Powerful for handling complex queries

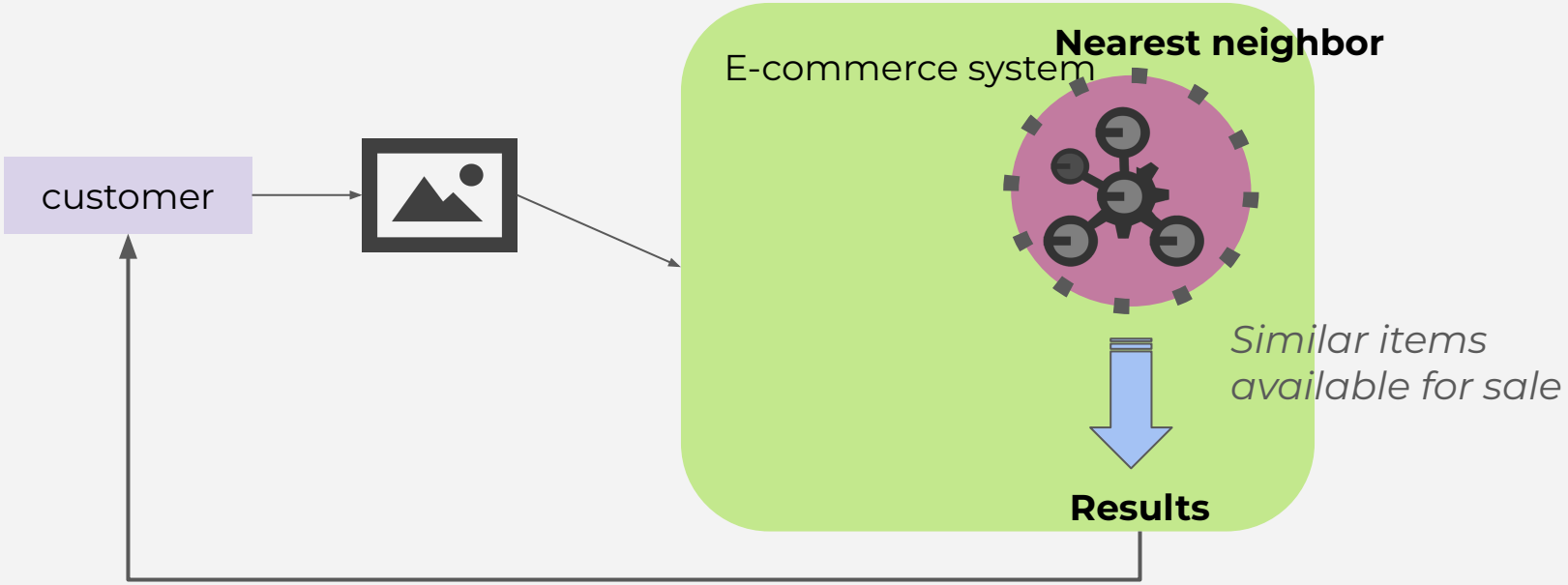
Here a 5 use cases:

1. Image Retrieval & Similarity Search
2. Recommendation Systems
3. Natural Language Processing (NPL)
4. Fraud Detection
5. Bioinformatics

Image Retrieval & Similarity Search

For example: ***an e-commerce platform***

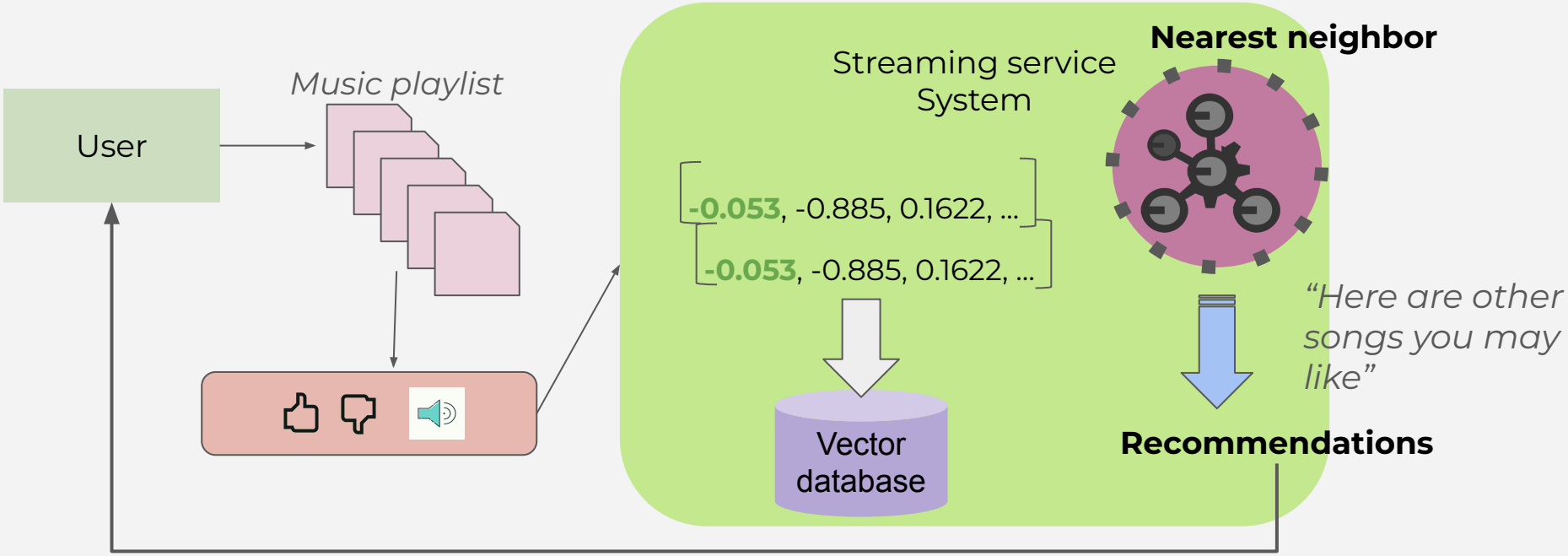
Can use a vector database for visual search feature.



Recommendation Systems

For example: ***music streaming service***

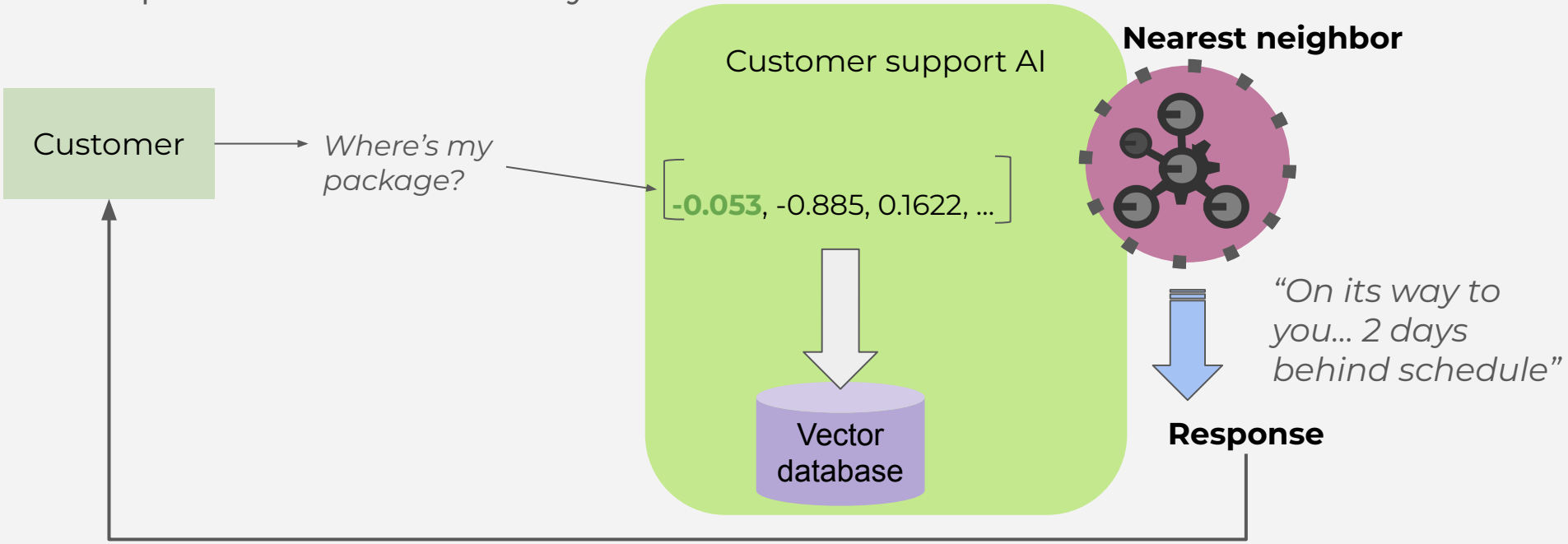
Uses a vector database to recommend songs



NLP - Natural Language Processing

For example: ***customer support AI***

Uses a vector database to understand and respond to user queries more efficiently



Fraud Detection & Bioinformatics

For example: ***detect fraudulent activities***

Uses a vector database to quickly compare user behavior patterns and flag anomalies...

For example: ***compare genetic sequences***

Uses a vector database to compare gene expression profiles from different patient samples.

LLM
Large Language Models

Large Language Models - what are they?

What Is a LLM?

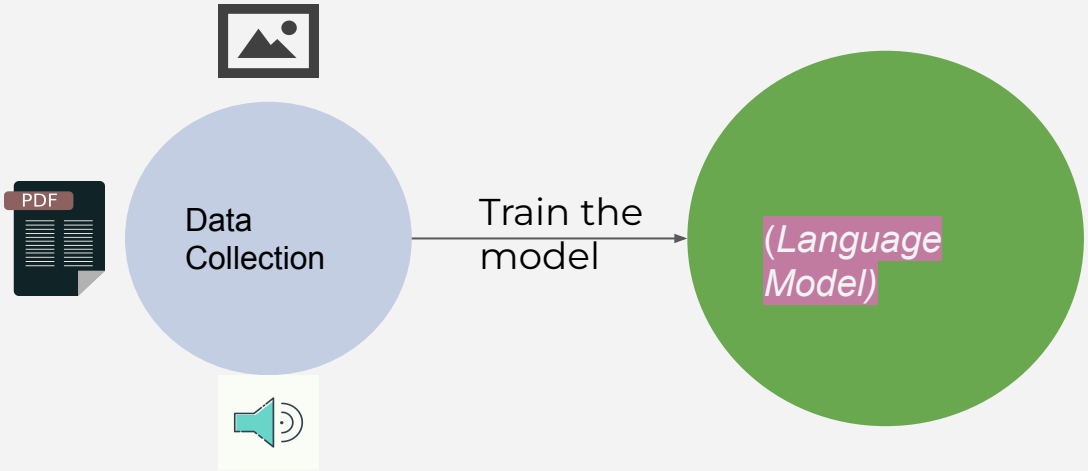
Large Language Model

A type of AI algorithm...

Trained on a large amount of text data

- Gives responses in natural language (*and other formats)*

Data Collection and Training...



Collection of lots of data & train the model: learn patterns in human language (*articles*, *studies*, *news*...)

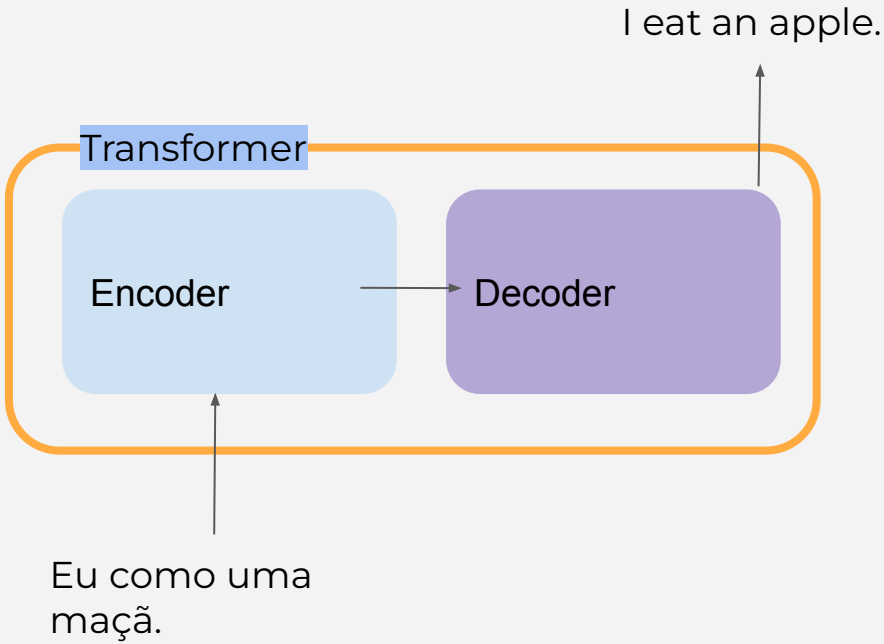
How LLMs are trained

There are a few steps involved in training an LLM:

- **Unsupervised learning** - the model starts to derive relationships between words and concepts, then fine tuned with supervised learning
- Next - training data goes through a **Transformer** - enabling the LLM to recognize relationships and connections using a self-attention mechanism

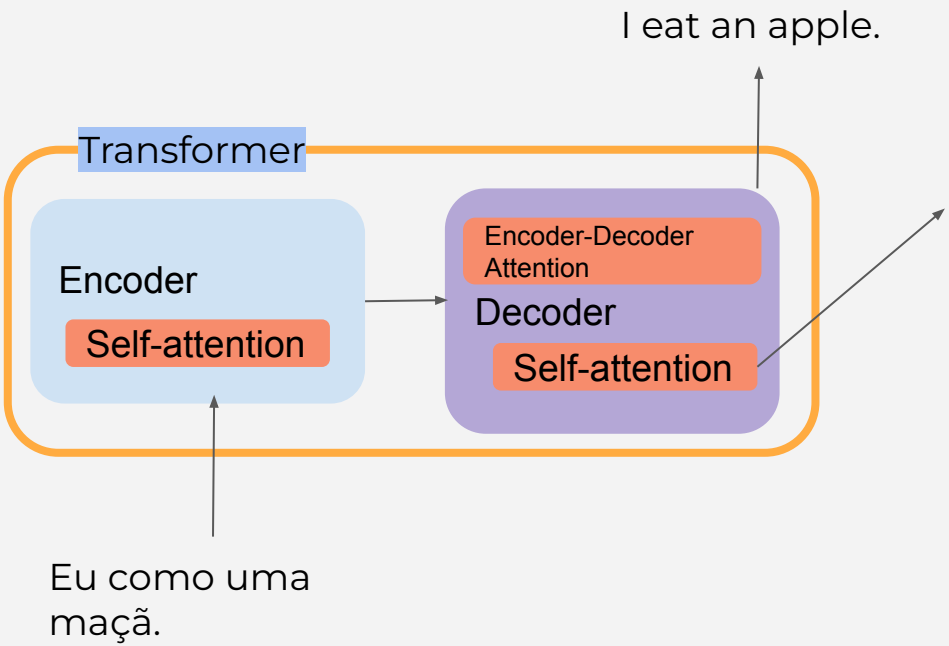
The Transformer Architecture - Overview

The **Transformer Architecture** is a *Neural Network* best suited for text and natural language processing.



GPT
(Language Model)

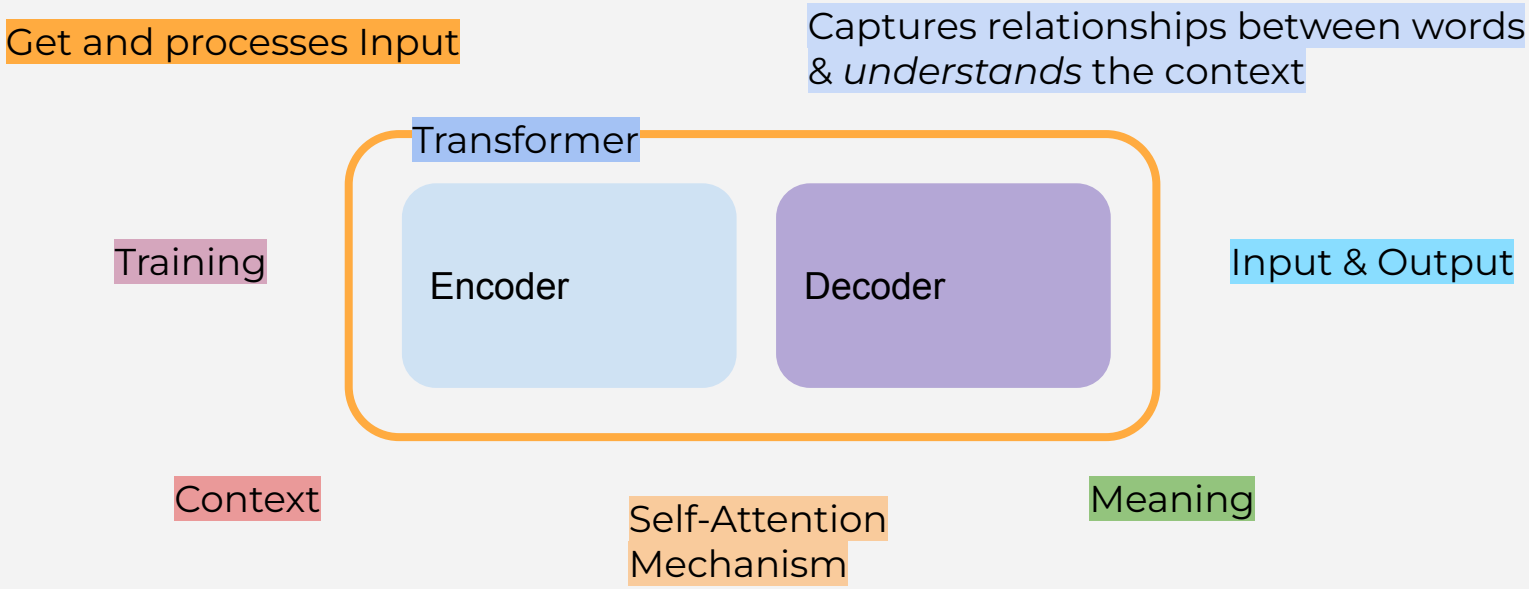
The Transformer Architecture - Overview



Self-Attention layer - decodes and looks at other words in the input... so to help lead to a better encoding for the words

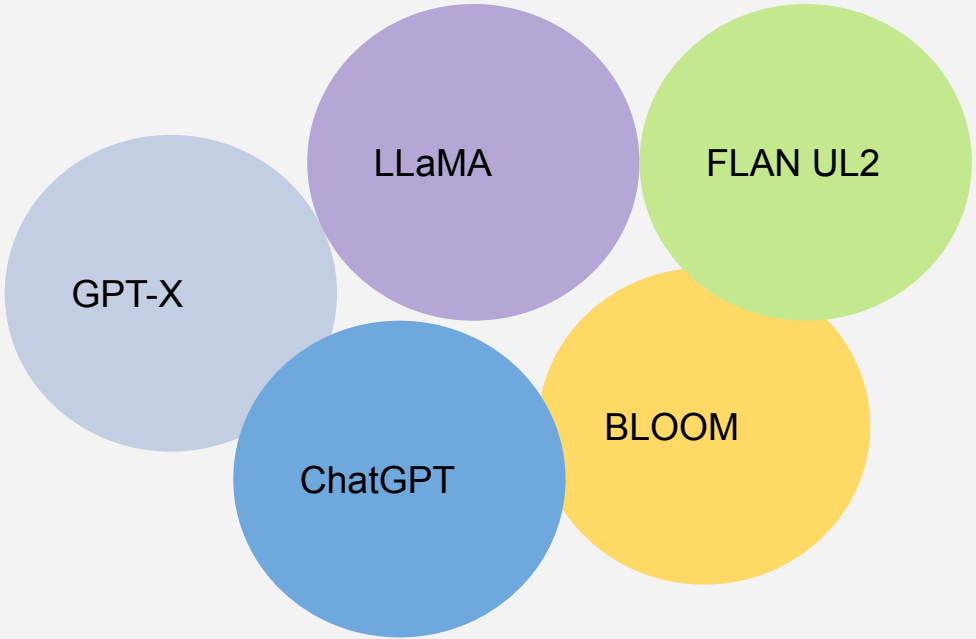
This helps de decoder focus on relevant parts of the input sentence.

The Transformer Architecture - Overview



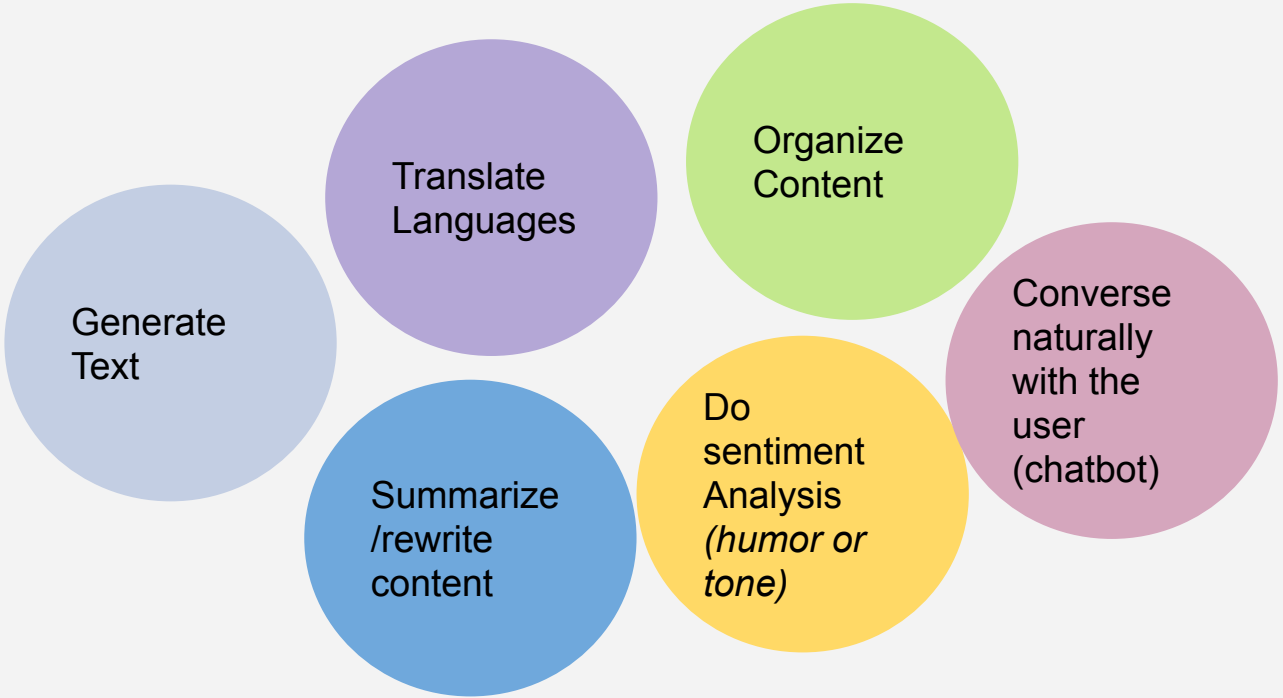
LLMs Available

To name a few...



LLMs have many use-cases

They can...

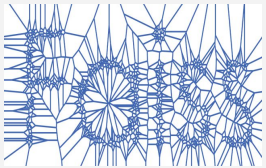


The top 5 Vector Databases

Pinecone

milvus

weaviate



Chroma

Pinecone

Key Features:

- Managed service - simple deployment and maintenance
- Supports real-time vector indexing & querying - scalability and performance

Unique Selling Points:

- Provides a simple API
- Strong focus on consistency

Milvus

Key Features:

- Open-source and designed for scalability and high-performance
- Supports both CPU and GPU

Unique Selling Points:

- Highly customizable
- Rich API and SDK (in multiple programming languages

Faiss (Facebook AI Similarity Search)

Key Features:

- Developed by Facebook - efficient similarity search
- Operates mainly in memory for fast data retrieval

Unique Selling Points:

- Provides highly optimized algorithms for similarity search
- Best suited for researchers and developers in AI

Weaviate

Key Features:

- Open-source vector engine - supports GraphQL, RESTful APIs...
- Has features like semantic search, automatic classification and object recognition

Unique Selling Points:

- Modular infrastructure
- Supports semantic search with a built-in knowledge graph

Annoy (Approximate Nearest Neighbors Oh Yeah)

Key Features:

- Lightweight, open-source library - fast
- High performance with memory-mapped files - good for large-scale datasets

Unique Selling Points:

- Prioritizes speed and memory efficiency
- Provides an easy-to-use interface with minimal setup

Chroma - AI Native & Open-source

Key Features:

- Embedding storage and search- allows embeddings and their a associated metadata
- Wide range support - various programming language integration
- Performance and scalability
- Open-source

Unique Selling Points:

- Simplicity and developer productivity
- High performance
- Customizable & extensible
- Cost-effective - free and open-source

Building Vector Databases

Hands-on - Building vector databases from scratch

Development Environment Set up

- VS Code
- Python
- OpenAI account and API Key

Hands-on - Create a Vector DB using Chroma

Vector database with Chroma

Chroma database workflow

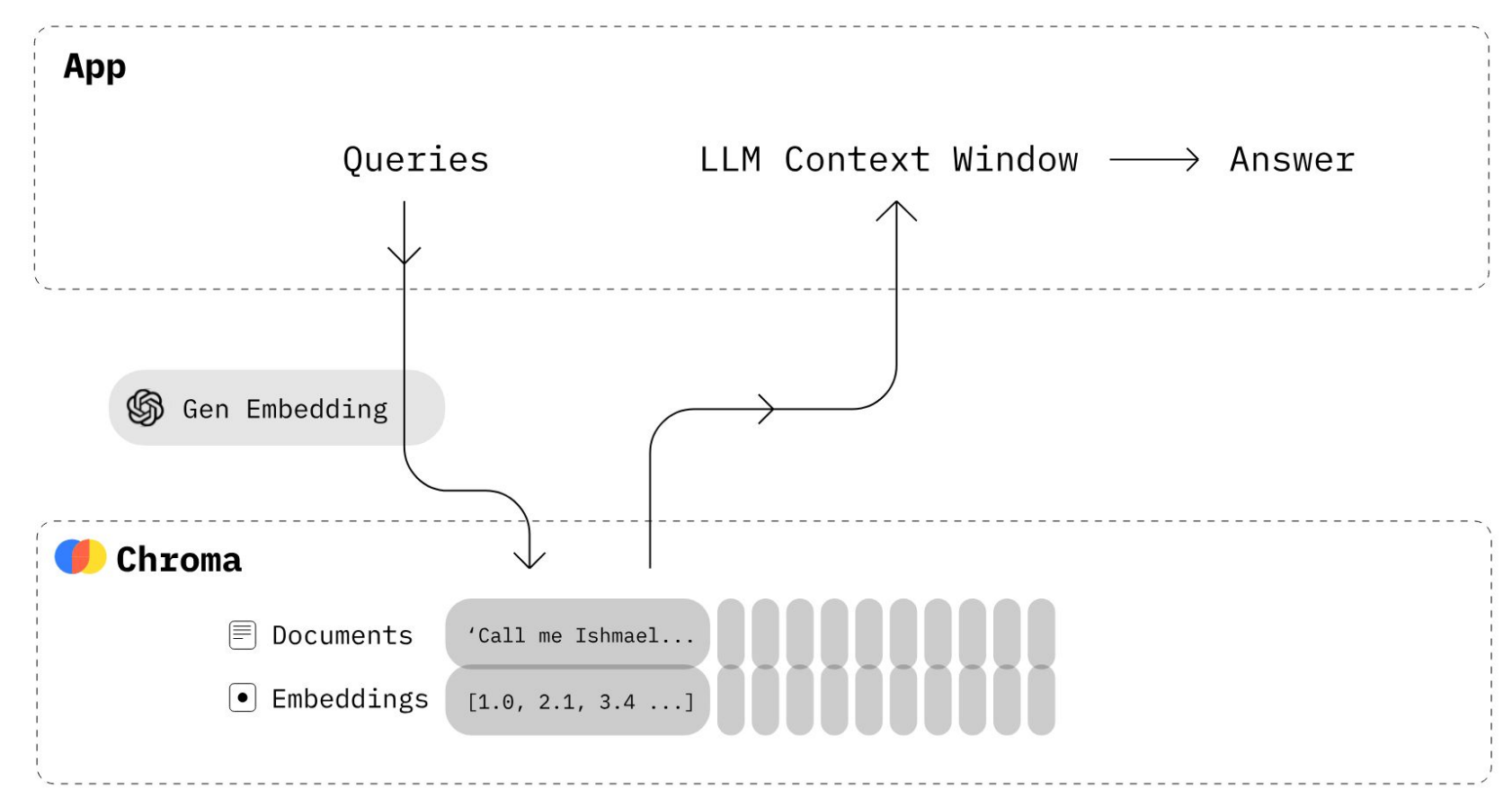
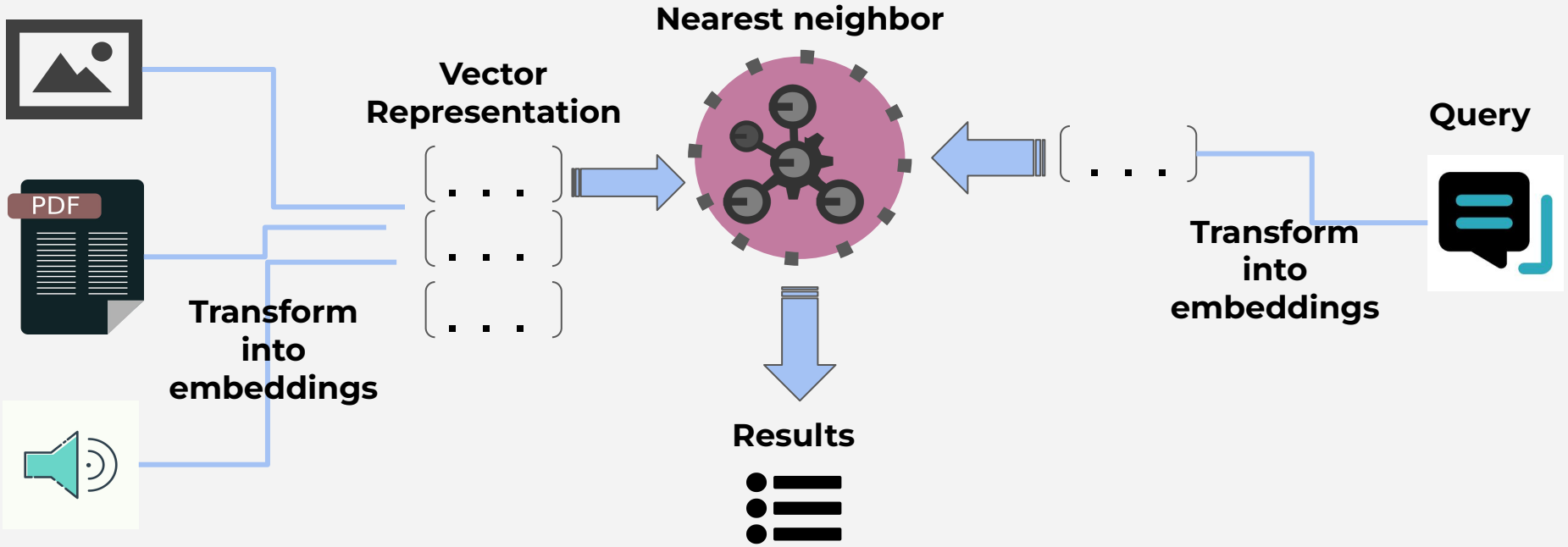


Image source:https://docs.trychroma.com/

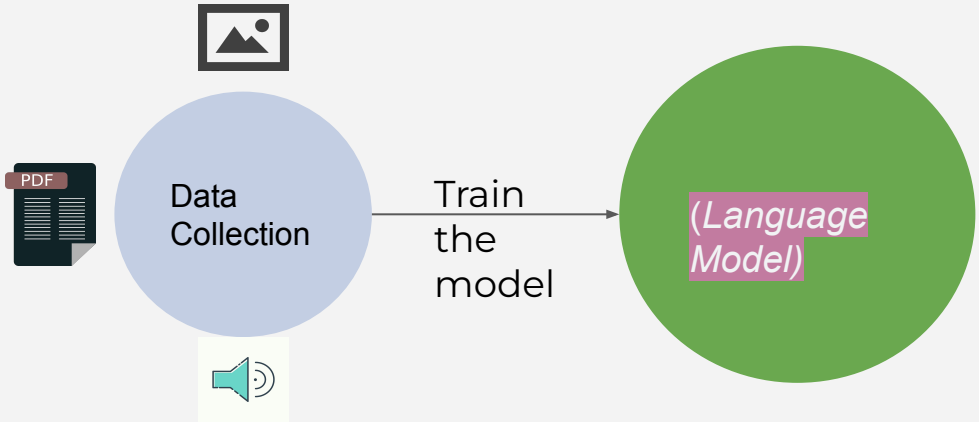
Vector Search - How Does It Work?



Chroma database - OpenAI Integration

Understanding Large Language Models (LLMs)

Data Collection and Training...



Collection of lots of data & train the model: learn patterns in human language (*articles*, *studies*, *news*...)

How Does it Work?

VectorStore holds **embeddings - Vector** representation of the text

But why ***embeddings***?

Because we can easily do search where we look for for pieces of text that are most similar in the vector space

- Capture semantic meaning
- Enabling similarity measures
- Handling high-dimension data

Vector Stores

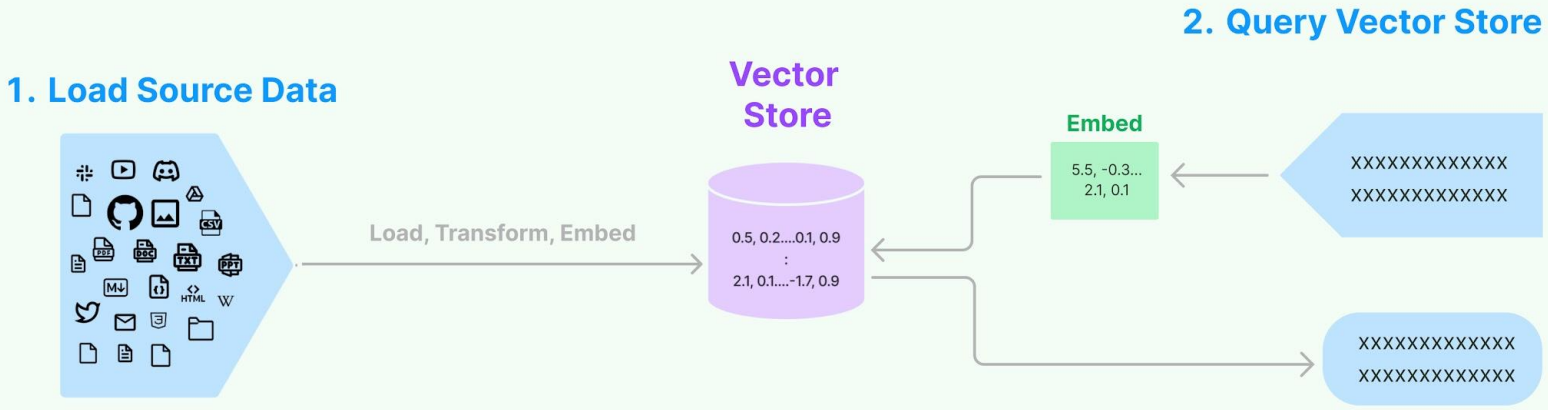
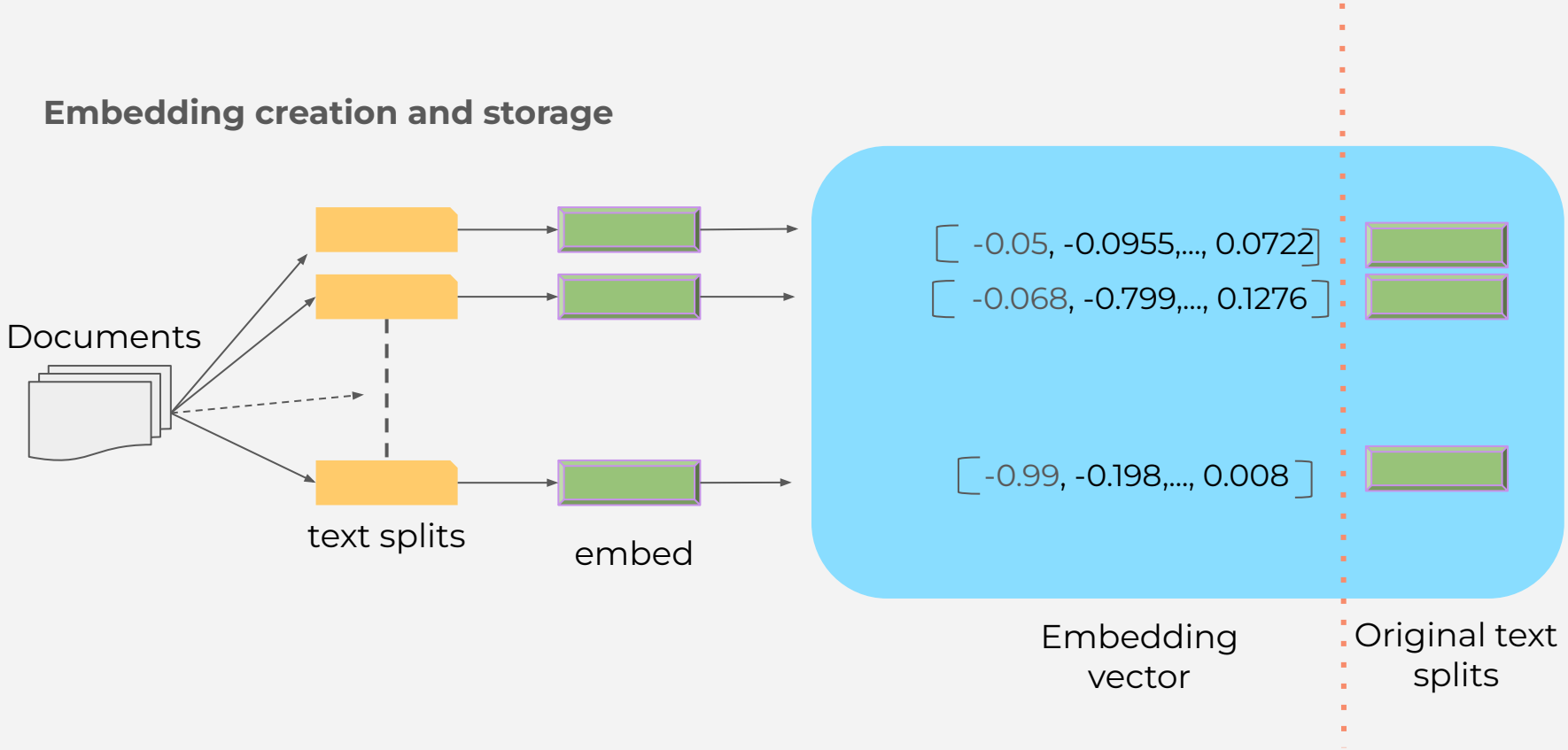
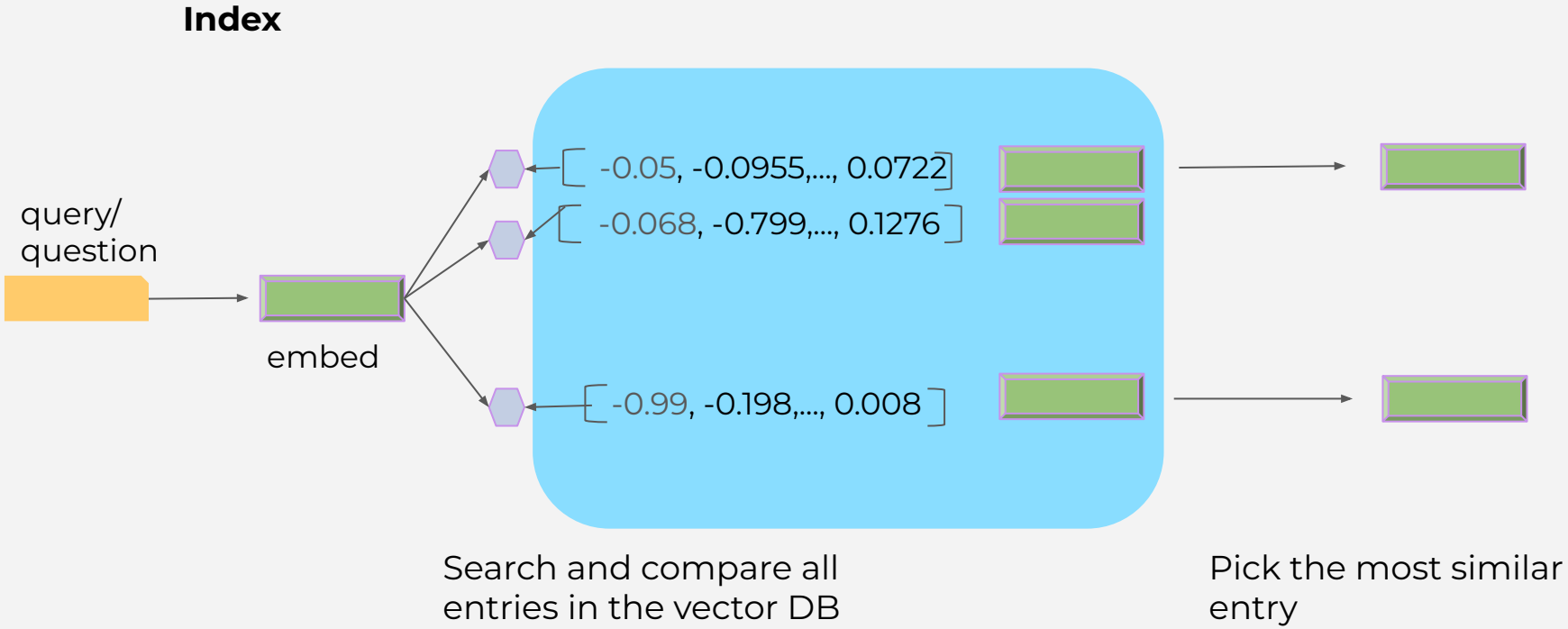


Image source: www.langchain.com

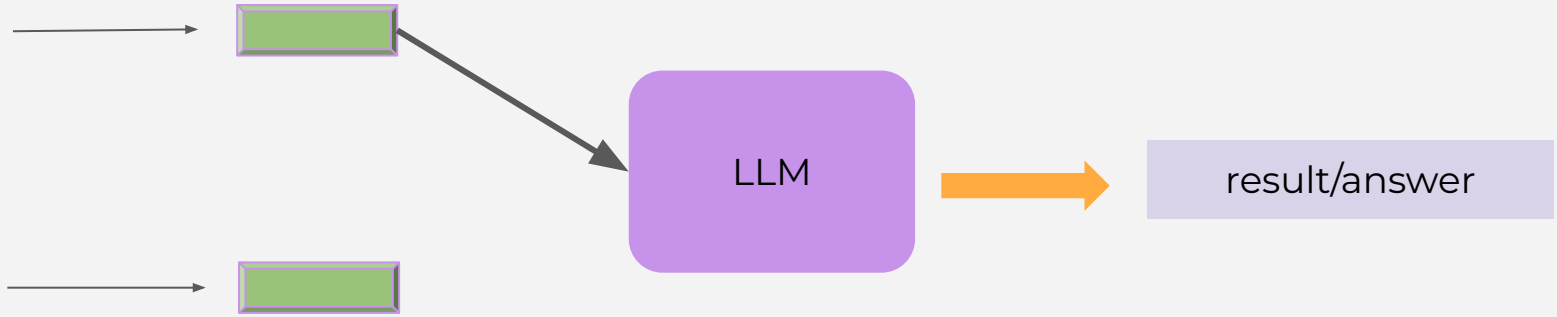
Vector database - full overview



Vector Database - querying the vector database



Vector Store - Processing with LLM

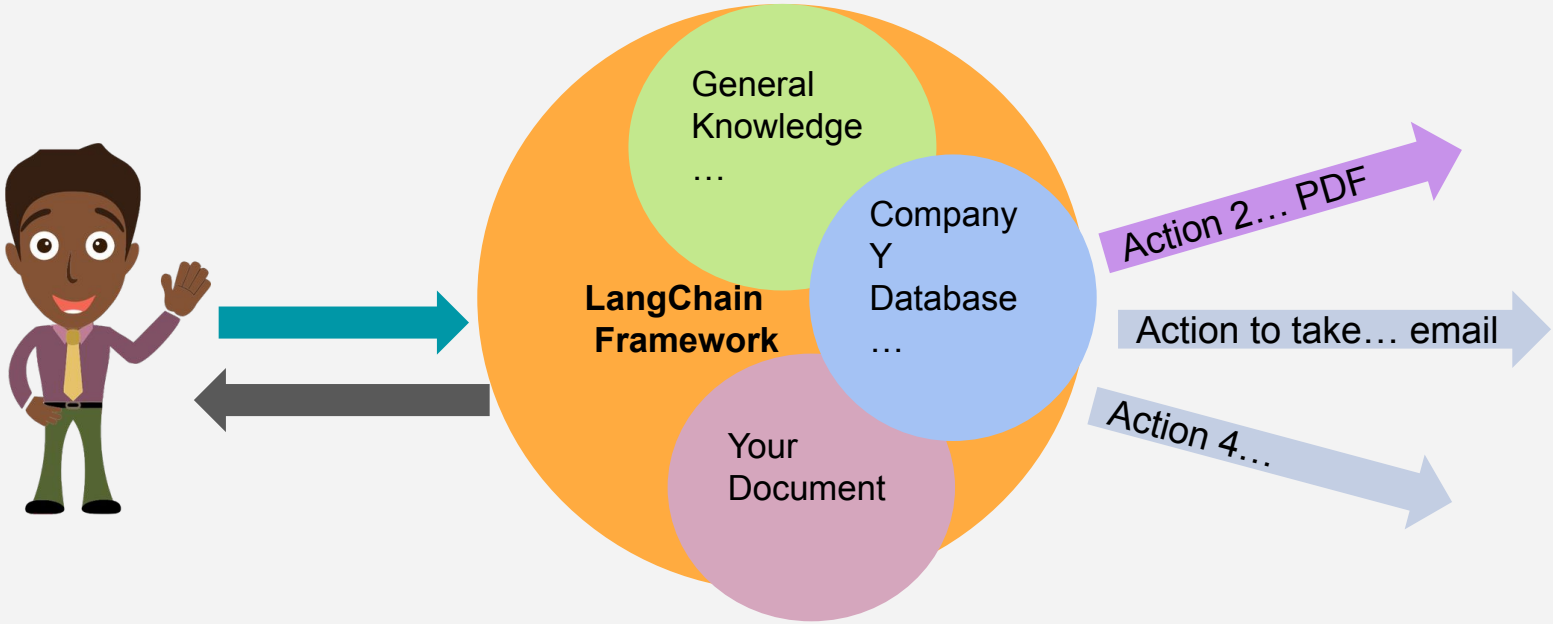


Take the most similar entry found and pass it on to the LLM along with the question/query...

Pick the most similar entry

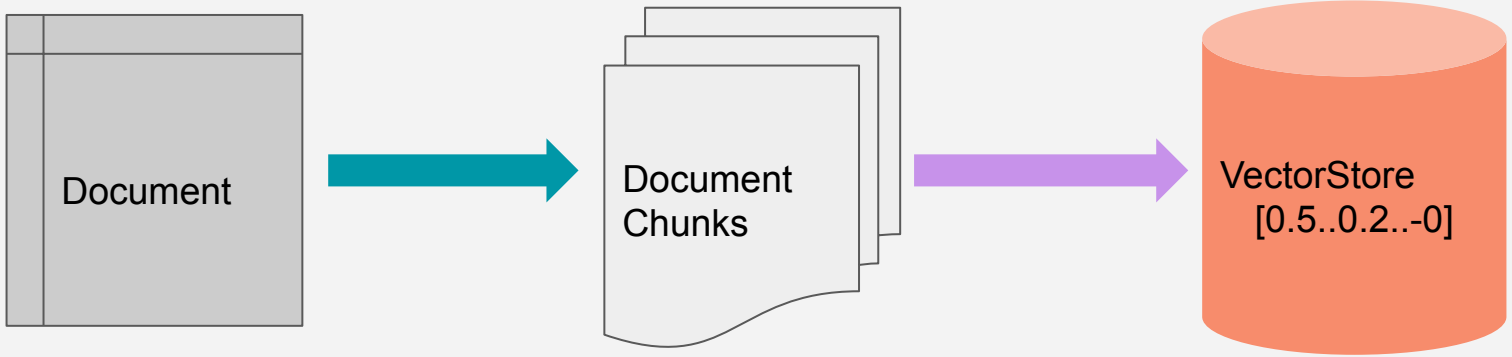
LangChain

A framework (*open source)* for building applications that leverage various LLMs (Large Language Models).
Additionally - you can also use **external sources of data** combined with various LLMs!



LangChain - How Does it Work?

The Framework takes a document and transforms it into VectorStore (stores the *chunks of data)*



Integrating OpenAI Embeddings with Chroma

Chroma Vector Database - Metrics and Data Structures

Metrics for Evaluating Vector Databases:

- **Latency:** time it takes to complete a query
- **Throughput:** # of queries that can be processed per unit of time.
- **Precision and Recall:** evaluation of accuracy of the search result
- **Memory Usage**
- **Scalability**

Chroma Vector Database - Metrics and Data Structures

Data Structures in Chroma

- Inverted Indexes
- K-d Trees
- Hierarchical Navigable Small World (HNSW) Graphs
- Locality-Sensitive Hashing (LSH)
- Priority Queues

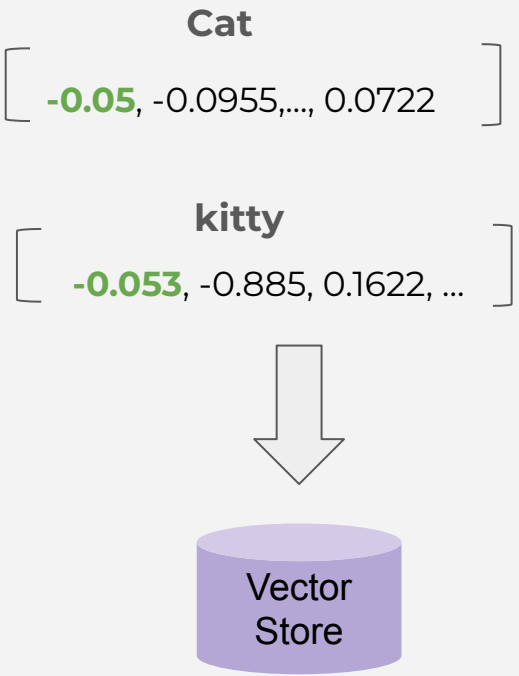
Metrics (Measuring Precision)- Hands on

- Set up a vector database with Chroma
- Insert data
- Perform queries
- Monitor key performance metrics

Measuring Throughput

- How many queries your system can handle per unit of time

Measuring Precision and Recall



Searching - *finding relevant results to the query string..*

Recommendations - *items with related text strings are recommended...*

Classification - *text strings are classified by most relevant and similar labels...*

Measuring Scalability

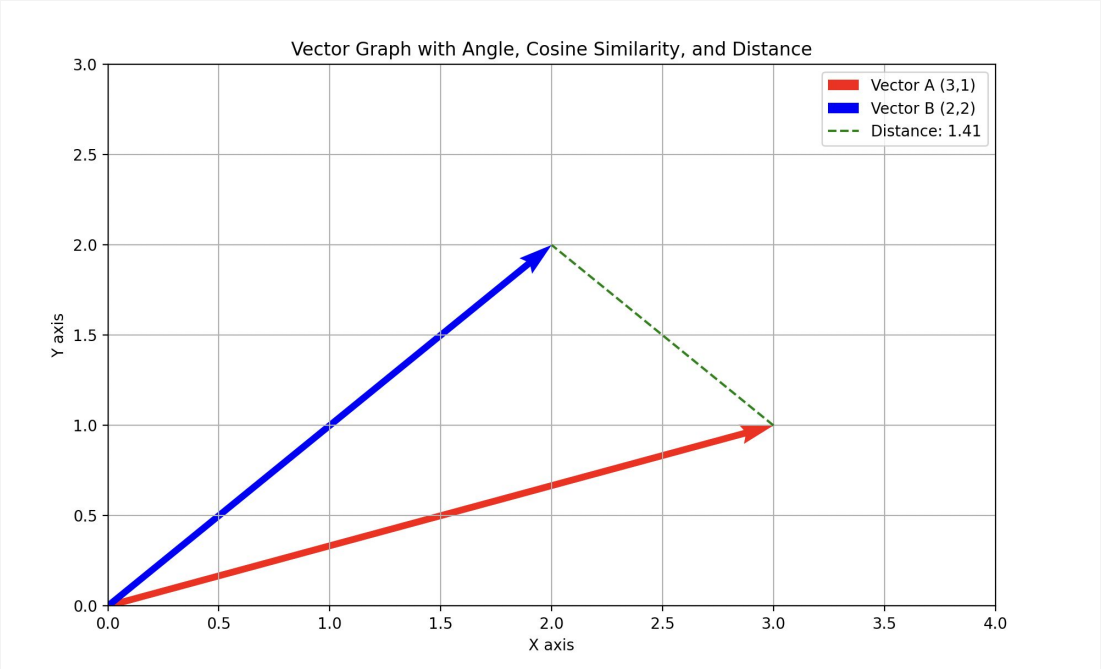
Create Vector Database with Pinecone

- Create account
- Dashboard overview
- Create a sample index
- Test it out

Use cases..

- Data Analysis (applications that can analyze large amounts of data...)
- Flight booking
- Study helper (learn material faster)
- Money transfer
- Code analyzer (debugging code, learn a large codebase fast)
- Personal AI assistants
- Connect to a variety of APIs (LLMs working with APIs through langchain)
- ...

Vector similarity - Deep dive



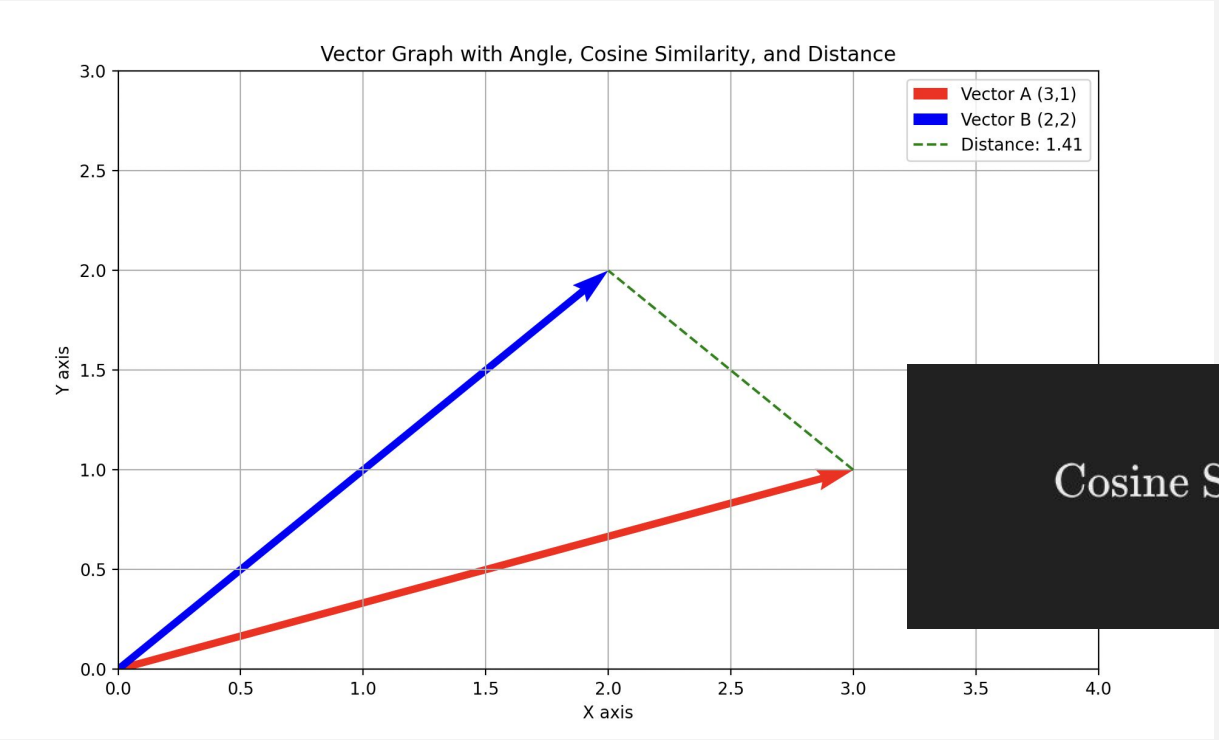
- A way to capture the closeness or alignment between two data points...
- Similarity influenced by
  - Direction
  - Magnitude
  - and relative position

Common Measures of Vector Similarity

- Cosine Similarity
- Euclidean Distance
- Dot Product

Common Measures of Vector Similarity

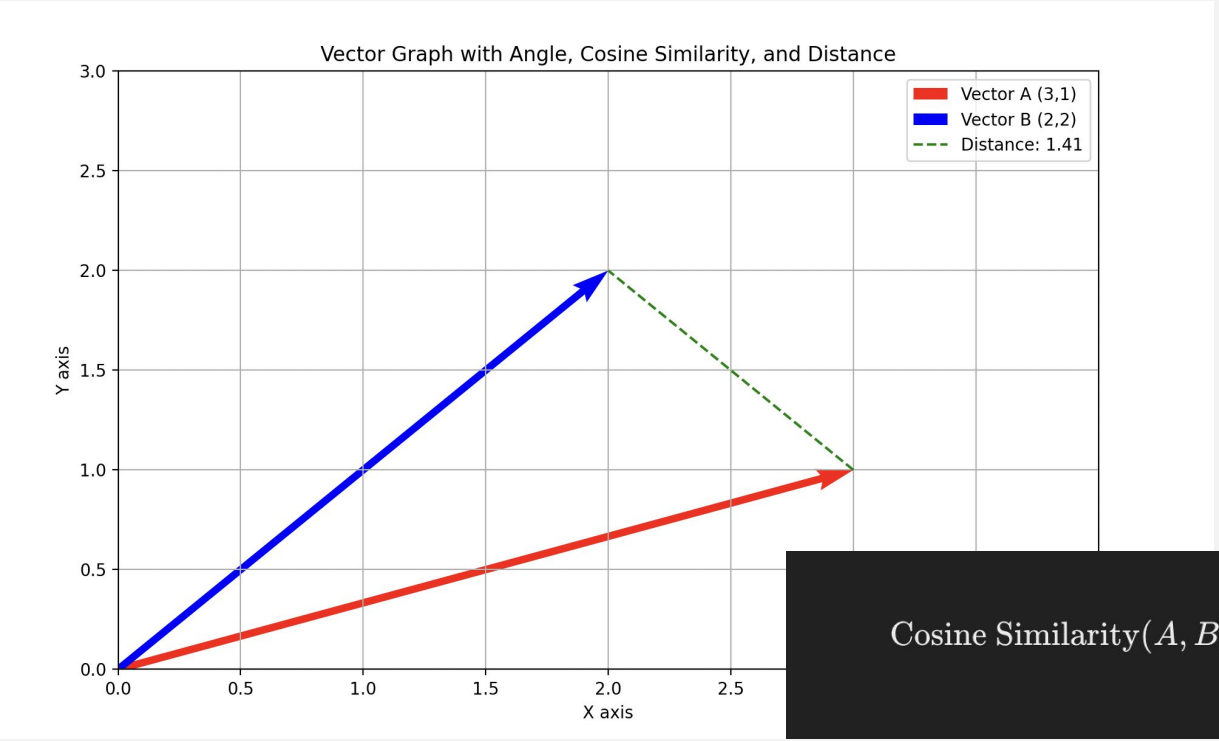
- Cosine Similarity



$$\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$$

Common Measures of Vector Similarity

- Cosine Similarity

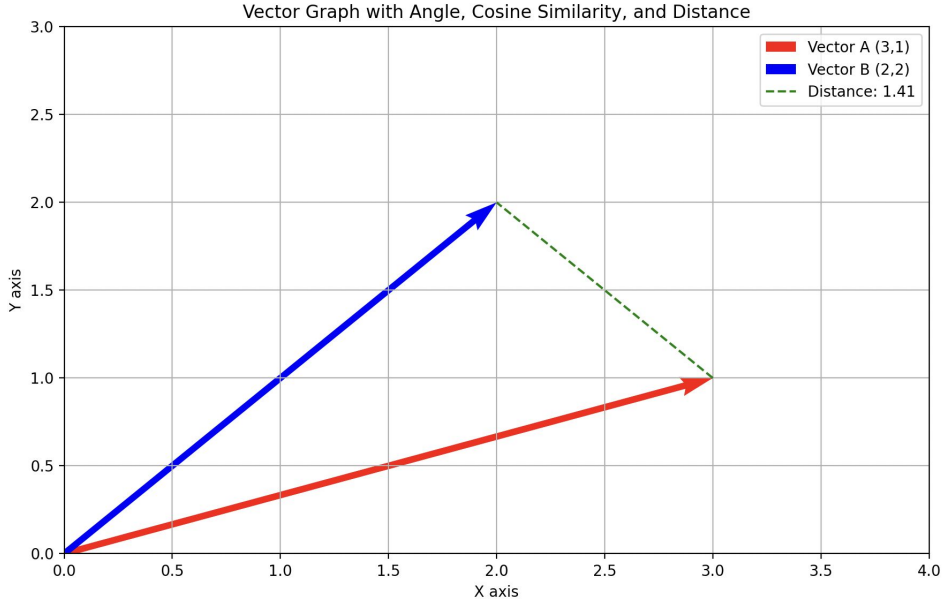


$$\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$$

- Magnitude (distance) doesn't matter
- The angle (cosine) is what matters
  - higher value means similar
  - *Values range between -1 and 1*

Common Measures of Vector Similarity

- Cosine Similarity



$$\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$$

- Dot prod For Arrow A and Arrow B, that's (3 * 2) + (1 * 2) = 6 + 2 = 8.
- Magnitude For Arrow A, For Arrow A, it's the square root of (3² + 1²) = √(9 + 1) = √10
- Magnitude For Arrow B, it's √(2² + 2²) = √(4 + 4) = √8.

- Cosine Similarity = $\frac{8}{\sqrt{10} * \sqrt{8}}$

Result: ~0.894

Common Measures of Vector Similarity

- Cosine Similarity

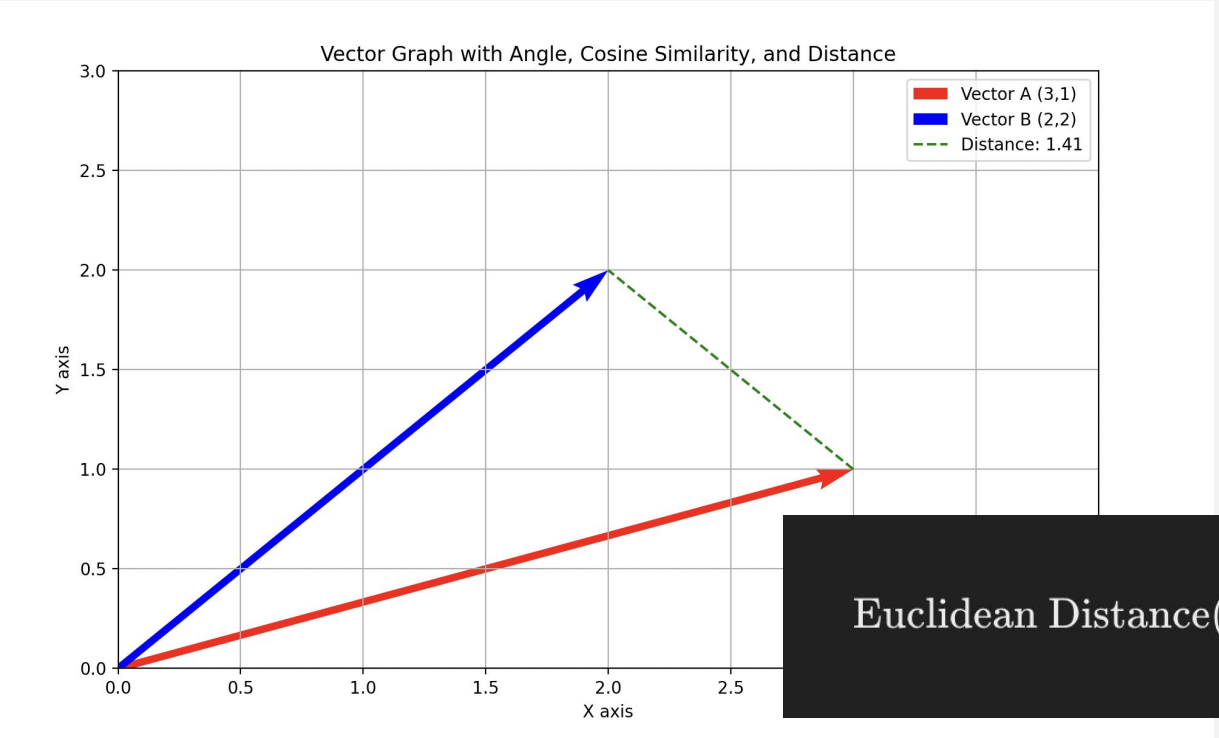
$$\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$$

Best use:

- Information retrieval and text mining

Common Measures of Vector Similarity

- Euclidean Distance / L2 Norm



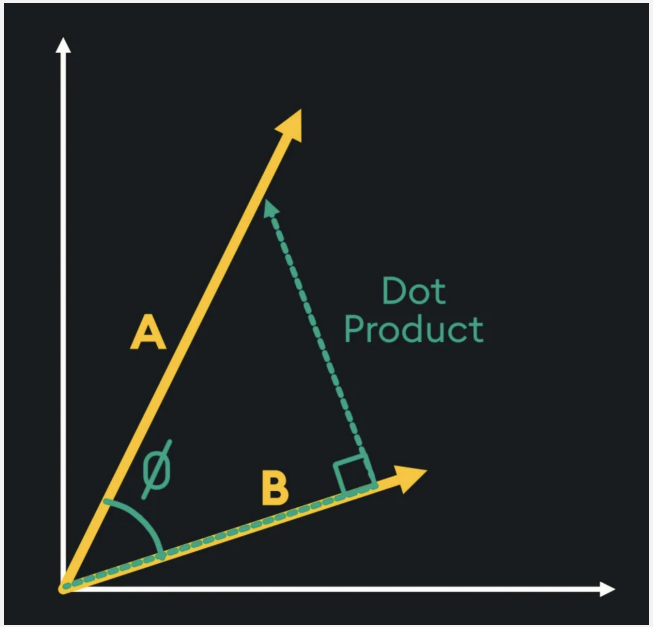
Best use:

- When vector magnitude need to be considered
- Ideal for clustering tasks (k-means clustering)

$$\text{Euclidean Distance}(A, B) = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2}$$

Common Measures of Vector Similarity

- Dot Product



Best use:

- Image retrieval & matching
- Music recommendation
- ...

$$\text{Dot Product}(A, B) = A \cdot B = \sum_{i=1}^{n} A_i B_i$$

Common Measures of Vector Similarity - Summary

$$\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$$

Best use:

- Topic modeling
- Document similarity
- Collaborative filtering

$$\text{Euclidean Distance}(A, B) = \sqrt{\sum_{i=1}^{n}(A_i - B_i)^2}$$

Best use:

- Clustering analysis
- Anomaly & fraud detection

$$\text{Dot Product}(A, B) = A \cdot B = \sum_{i=1}^{n} A_i B_i$$

Best use:

- Image retrieval & matching
- Neural networks & Deep Learning
- Music Recommendation

Hands on - Real world use case

Best use:

- Vectorization of data
  - Split large text/documents
  - Create embeddings
  - Save them to vector database

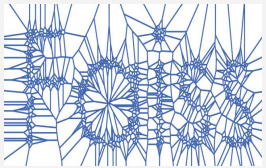
- Query (similarity search)

The top 5 Vector Databases

Pinecone

milvus

weaviate



Chroma

Pinecone

Key Features:

- Managed service - simple deployment and maintenance
- Supports real-time vector indexing & querying - scalability and performance

Unique Selling Points:

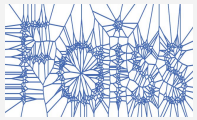
- Provides a simple API
- Strong focus on consistency

Create Vector Database with Pinecone

- Create account
- Dashboard overview
- Create a sample index
- Test it out

The top 5 Vector Databases

weaviate



Challenge:

- Explore the other vector databases:
  - Weviate
  - Faiss
  - Milvus
  - ...

Pinecone Summary

- Introduction to Pinecone
- Pinecone basics and set up
- Used LangChain Framework
- Attached an LLM to complete the workflow

Comparison of Vector Databases Deployment options

Vector Database	Local Deployment	Cloud Deployment	On-premises Deployment
Pinecone	❌	✅ (managed)	❌
Milvus	✅	✅(self-hosted)	✅
Chroma	✅	✅(self-hosted)	✅
Weaviate	✅	✅(self-hosted)	✅
Faiss	✅	❌	✅

Comparison of Vector Databases - Integration and API

Vector Database	Language SDK	REST API	GRPC API
Pinecone	Python, Node.js, Go, Rust	✅	✅
Milvus	Python, Java, Go, C++, Node.js, RESTful	✅	✅
Chroma	Python	✅	❌
Weaviate	Python, Java, JavaScript, .NET	✅	✅
Faiss	C++, Python	❌	✅

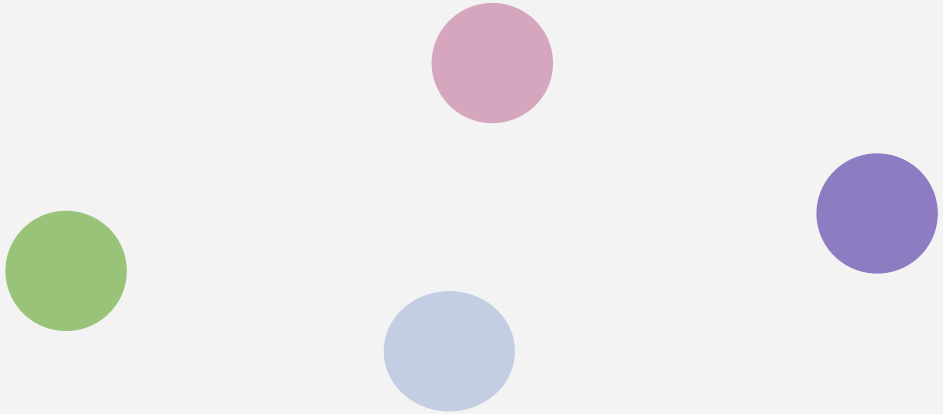
Community and Ecosystem

Vector Database	Open-source	Community	Integration with Frameworks
Pinecone	❌	✅	✅
Milvus	✅	✅	✅
Chroma	✅	✅	✅
Weaviate	✅	✅	✅
Faiss	✅	✅	✅

Pricing

Vector Database	Free Tier	Pay-as-you-go	Enterprise Plans
Pinecone	✅	✅	✅
Milvus	✅	❌	❌
Chroma	✅	❌	❌
Weaviate	✅	❌	✅
Faiss	✅	❌	❌

Which Vector Database Should I Use?



It depends on... what you what to accomplish. And other many factors...

1. **Project Requirements**
    a. Scalability
    b. Performance needs
    c. Data type
2. **Ease of Use and Integration**
    a. Developer experience
    b. Community support
3. **Feature Set**
    a. Advanced features
    b. Customization and flexibility
4. **Cost and Infrastructure**
    a. Budget constraints
    b. Infrastructure Needs
5. **Security and Compliance**
    a. Data security

Recommended Approach

To determine the best fit:

- **Evaluate a shortlist** - based on discussed criteria, shortlist a few databases
- **Prototype**
- **Community feedback**

Choosing the Right Vector Database

- **Data Type and Volume**
  - Text, Images or Audio: Weaviate or Chroma
  - Scalability: vertical and horizontal - Pinecone and Milvus
- **Query Performance and Latency**
  - Latency requirements - low latency applications - Pinecone (recommendation systems or live content filtering)
- **Accuracy and Precision**
  - Metric support - supports similarity metrics (cosine, euclidean...)
  - Tuning capabilities
- **Ease of Integration and Use**
  - API and Client Libraries
  - Documentation and Community
- **Cost considerations**
  - Pricing Structure (pay-as-you-go...)
  - Total cost of ownership
- **Security and Compliance**
  - Data Security
  - Compliance - DDPR, HIPAA
- **Vendor Stability and Support**
  - Vendor Reputation
  - Support Services

Congratulations!

You made it to the end!

- Next steps...

Course Summary

- Foundations of Vector Databases
  - What are they?
  - What problem vector databases solve?
  - Top 5 vector database
  - Key Differences
  - Challenges and use cases
  - How to build vector databases from scratch
    - Metrics and data structure
    - Vectorization with abstraction frameworks
  - Hands-on use cases: full AI-based application workflow (with LLMs)
  - Vector database comparisons
  - How to choose a vector database

Wrap up - Where to Go From Here?

- Keep learning
  - Get more ideas and build more applications and test different, less known vector databases!

- Read documentation - That’s where the Gold Is!
- Challenge yourself to keep learning new skills!

Thank you!

Traditional vs Vector Databases - Summary

- Traditional vs Vector databases
  - Limitations and Contrasts

- Vector databases & Embeddings - Overview
- How vector database work & advantages
- Vector databases Use cases

Building vector database - Hands-on

- Set up development environment
  - VS Code
  - Python
  - OpenAI API
- Chroma database workflow overview
- Creating a chroma database
- Default embedding function
- Creating OpenAI embeddings
- Vector database metrics