Vector Databases

as an Instrument of Big Data Analysis

Evgeniya Sukhodolskaya,
Developer Advocate,
Qdrant

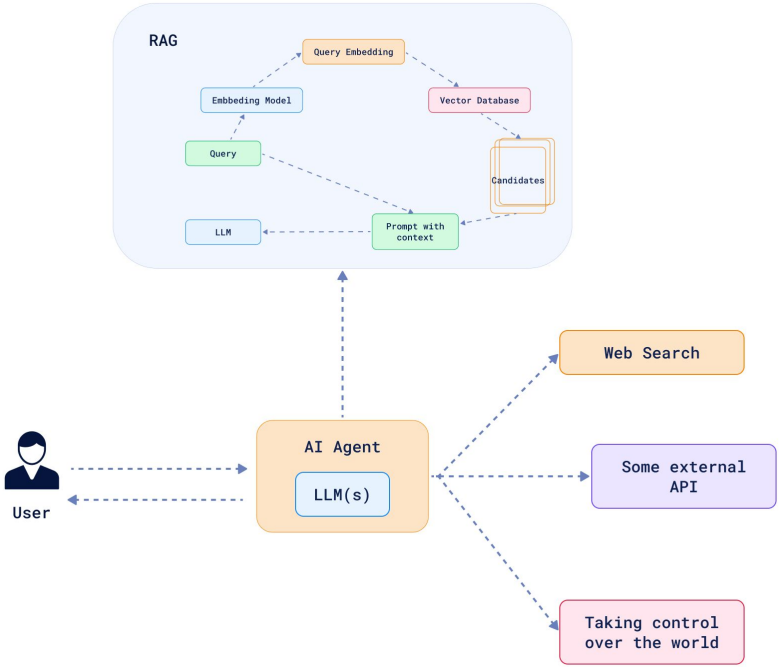
As a knowledge base

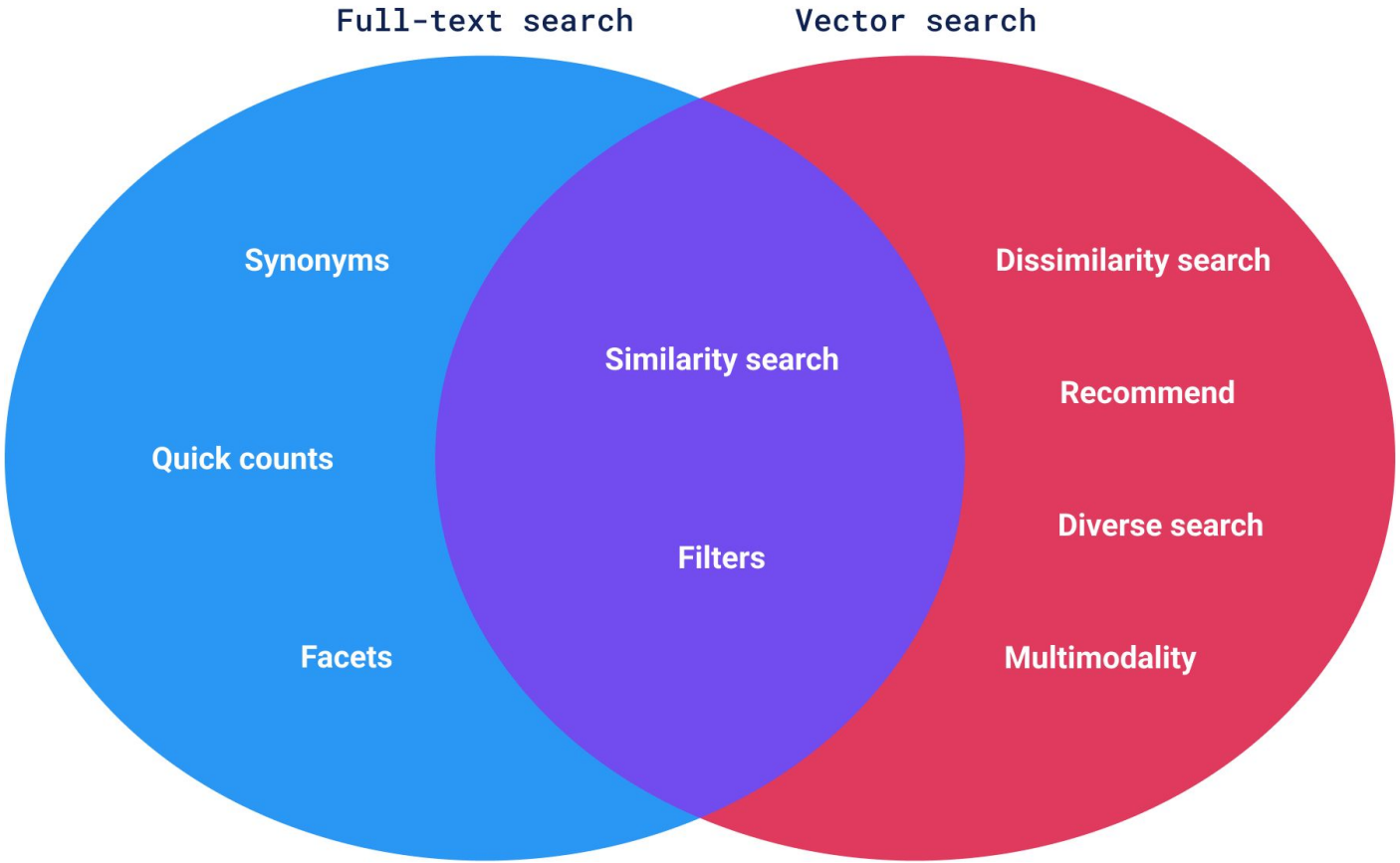
Using **semantic similarity** search
or **hybrid** search on **different data modalities**

For example, **Agentic RAG**

Or **MCP-based RAG solutions**
(f.e., as a support for vibe coding)

mcp-server-qdrant Public




Full-text search
Vector search
Synonyms
Quick counts
Facets
Similarity search
Filters
Dissimilarity search
Recommend
Diverse search
Multimodality

Agenda

- **Visual Data Analysis** with Vector Search;
- Vector Search for **Scalable Data Analysis**;
- Vector Search-Based Data Analysis as a **Business Solution**.

Why even?

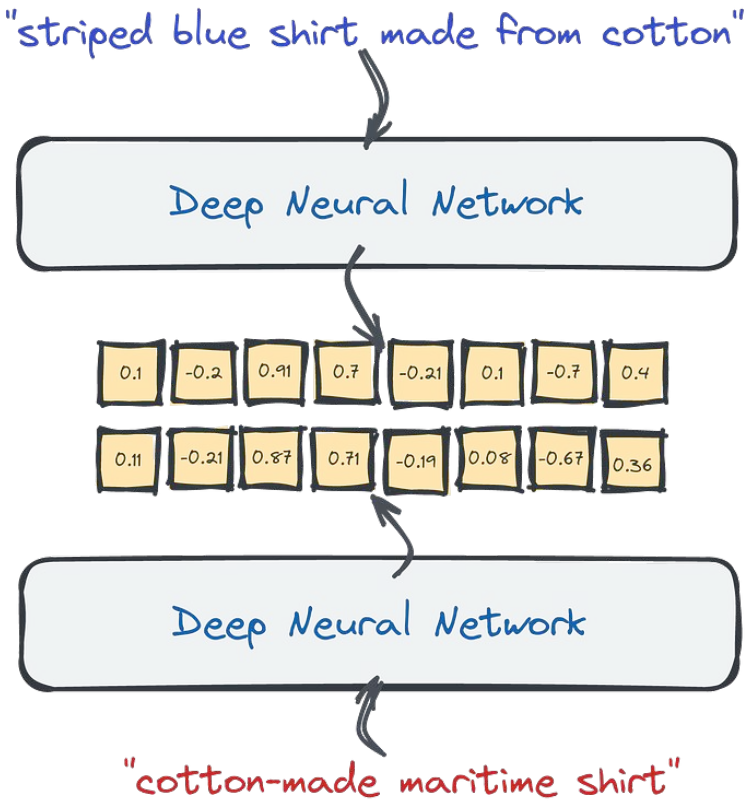
- I don’t know my data & I need to know it;
- There is a lot of it — like **billions**;
- I don’t have much **labeled data** to train automated analysis methods;
  - Or I’m not sure about the **quality** of my labeled data;
- Or those automated analysis methods are **heavy & slow;**
  - And/or **costly** (GPT-4o, please, analyze my billion points...);
- For what’s worse — **data shifts/drifts** in **production**, dynamically changing ALL the time;
- AAAAA!

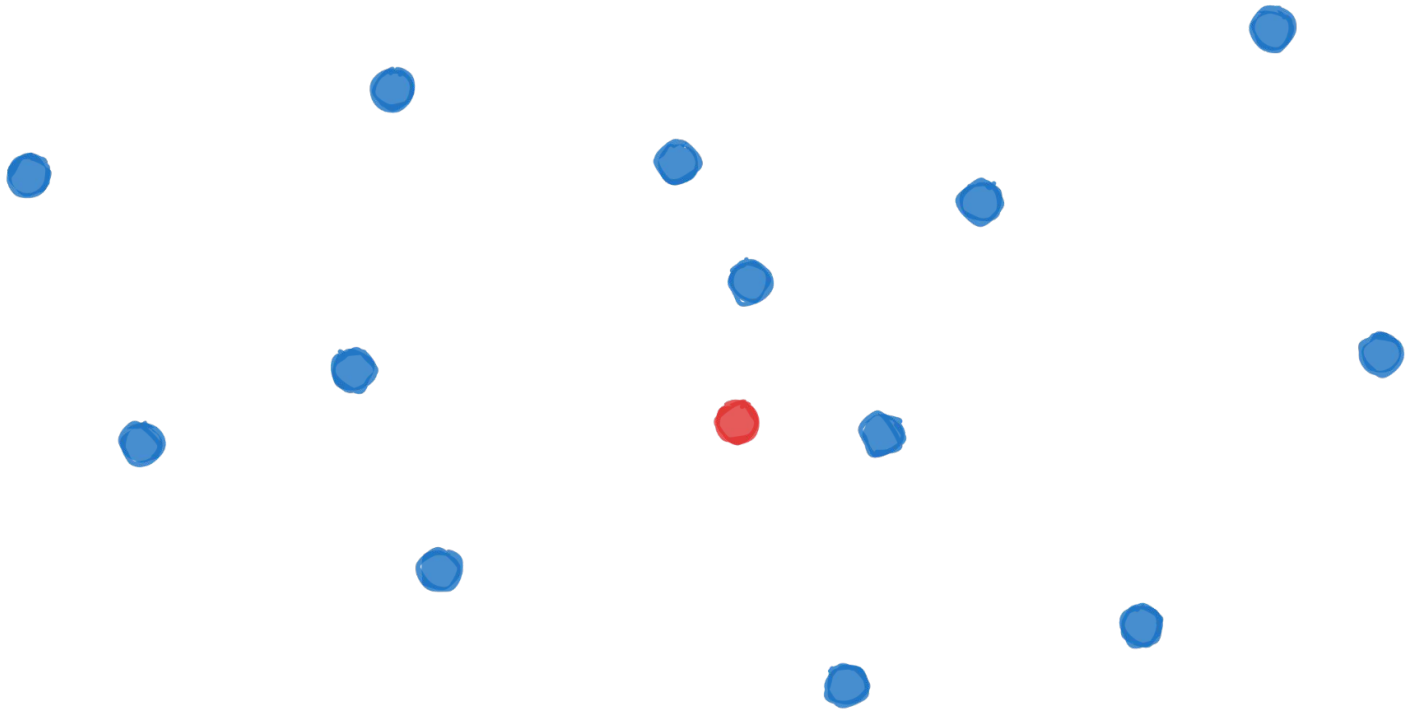
So, hear me out: Vector Search Solutions for Data Analysis.

Vector Search Basics for Unstructured Data Analysis

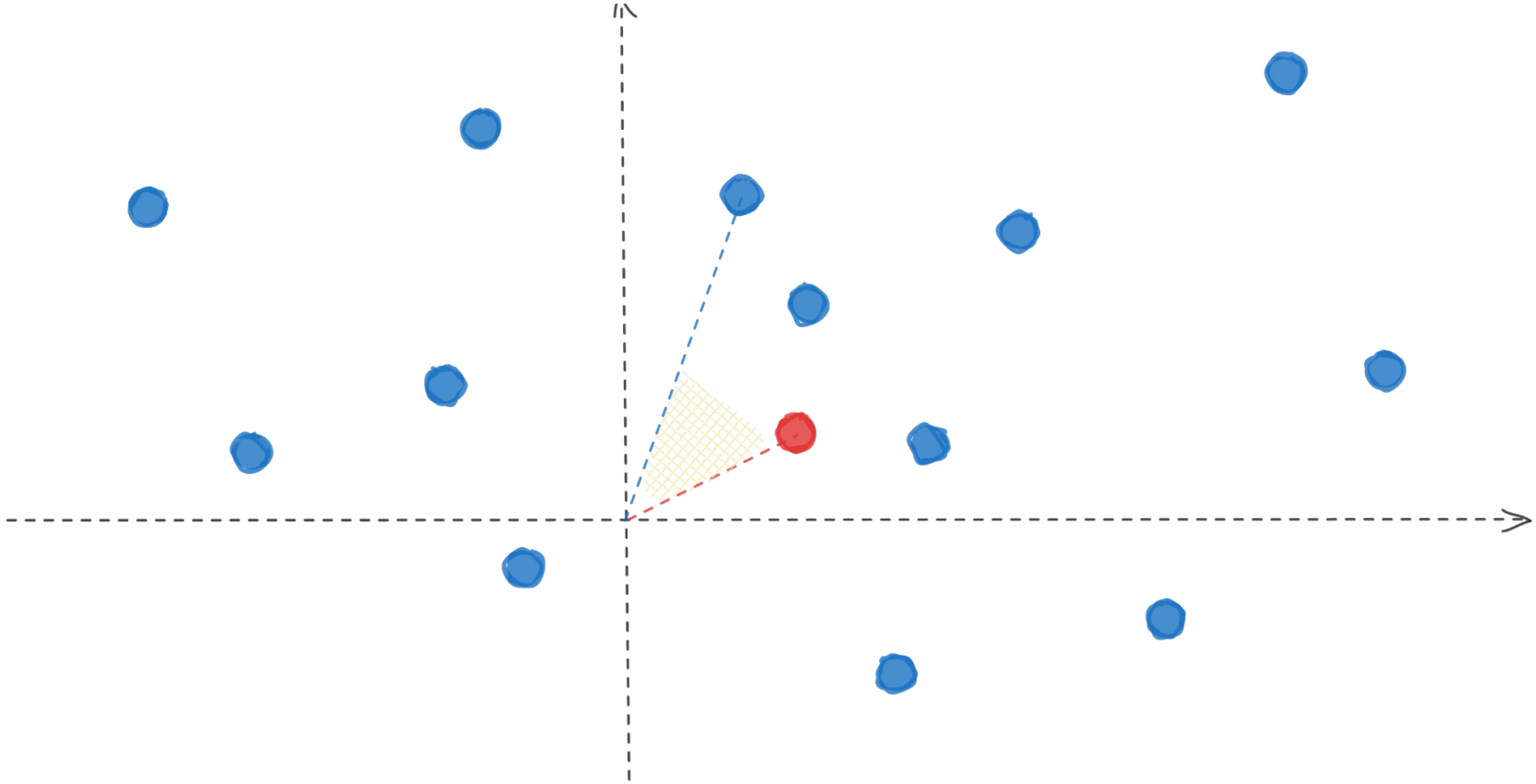
Basics of vector search

Data points are being modelled as fixed-dimensional vectors, created through some neural encoders.

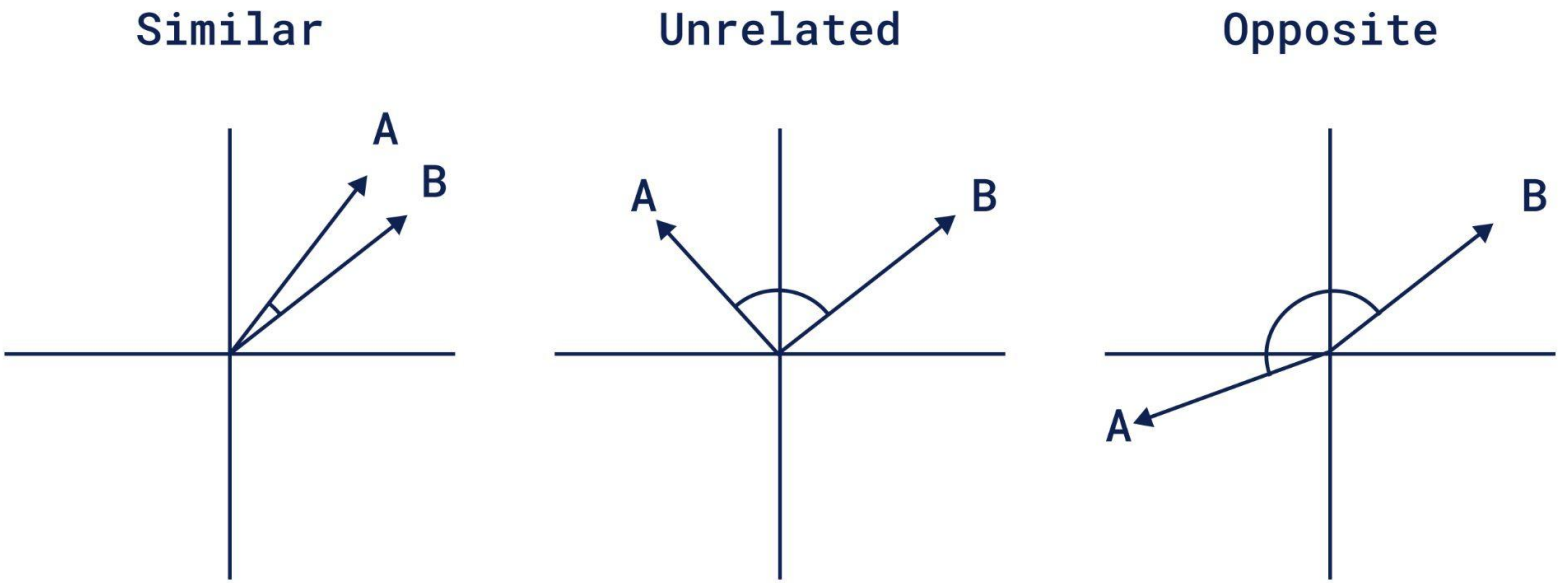




A distribution of data points vectors in 2d space.



Calculating the **cosine similarity** is based on the angle between two vectors.



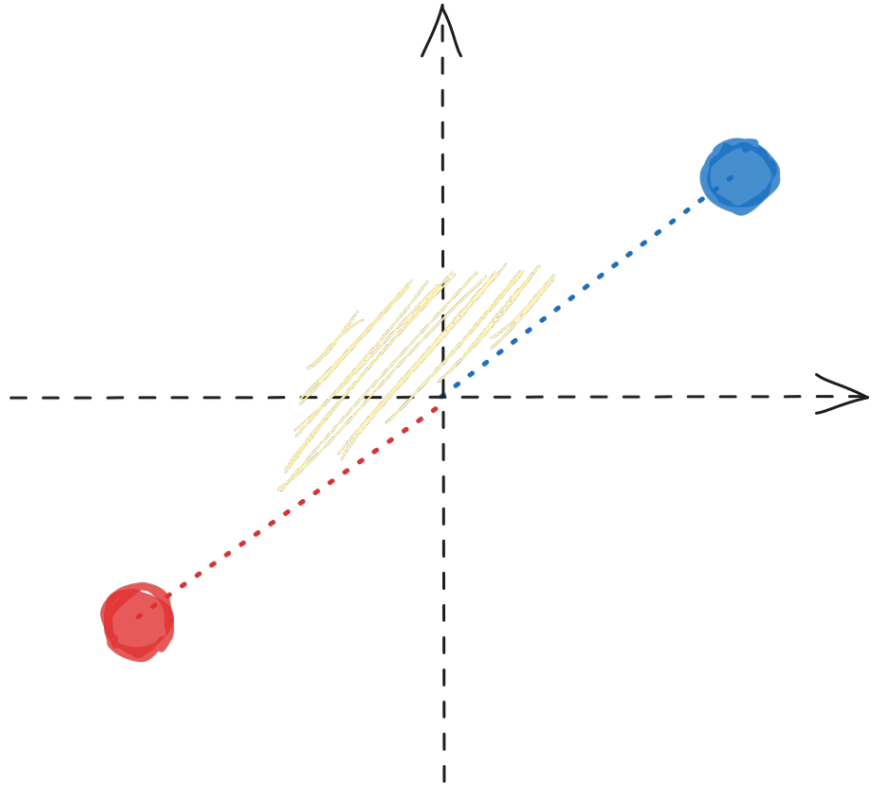
The closest points will have the highest cosine similarity score (a value of 1 means a **perfect match**, while -1 is the **opposite direction vector**).

Introducing DISsimilarity Search

The cosine similarity of a vector with its opposite is always -1.

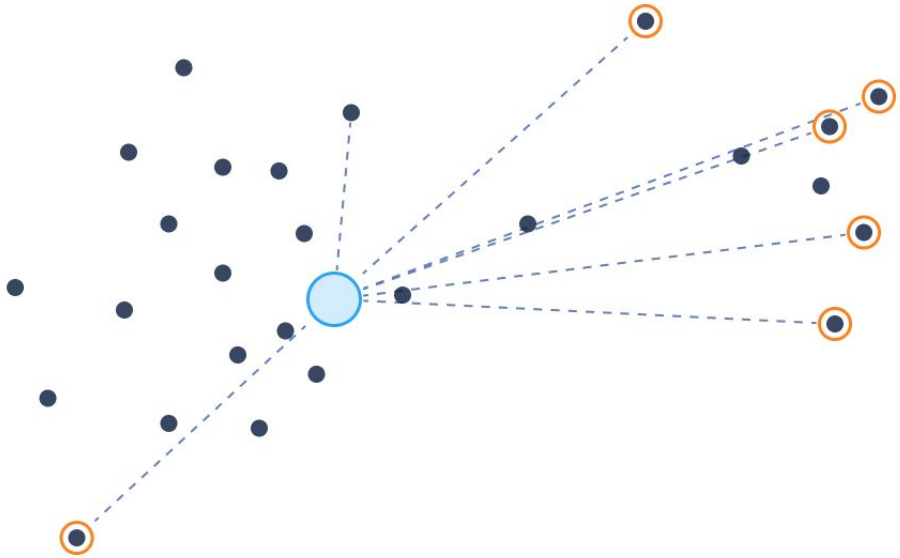
So, if we multiply a data point vector by -1, we get the complete opposite.

And the most similar point to this opposite = **the most DISsimilar data point** to our initial example.



Dissimilarity search

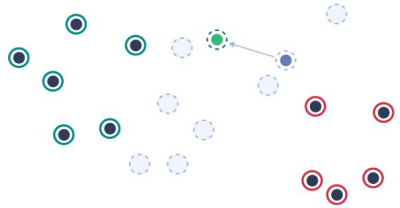
Finding the furthest points from the given query vector



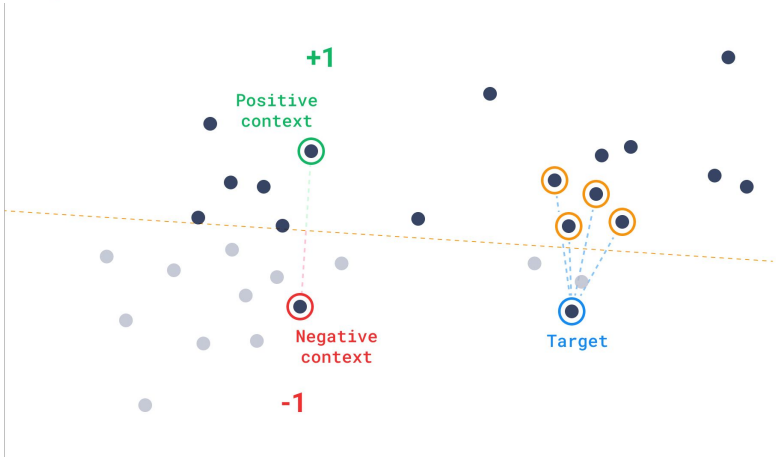
Qdrant’s Abilities

Beyond (dis)similarity search:

- Distance matrix API
- Query Points API:
  - Recommendations
    - “best_score” strategy
    - “average_vector” strategy
  - Discovery
    - Context Search
    - Discovery Search
  - Random Sampling
  - …



	Vec 1	Vec 2	Vec 3	Vec 4	Vec 5	Vec 6
Vec 1	-0.25	0.01	0.56	0.03	-0.39	-0.24
Vec 2	0.01	0.73	-0.19	-0.0	-0.19	0.14
Vec 3	0.56	-0.19	-0.64	-0.2	-0.5	-0.41
Vec 4	0.03	-0.0	-0.2	-0.72	-0.19	0.32
Vec 5	-0.39	-0.19	-0.5	-0.19	0.18	0.01
Vec 6	-0.24	0.14	-0.41	0.32	0.01	0.62

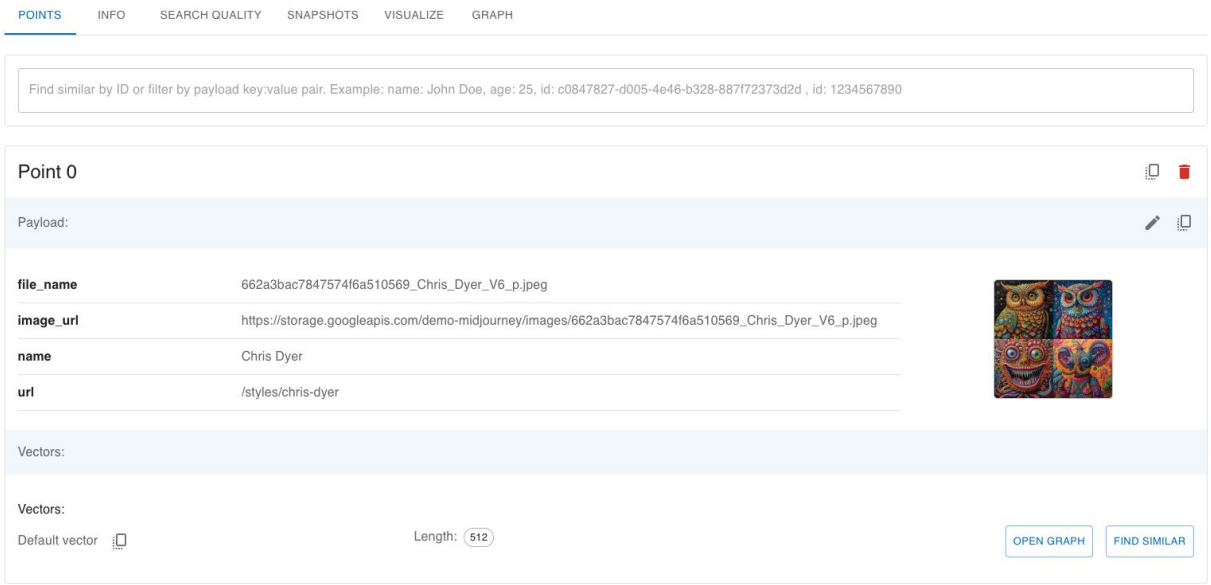


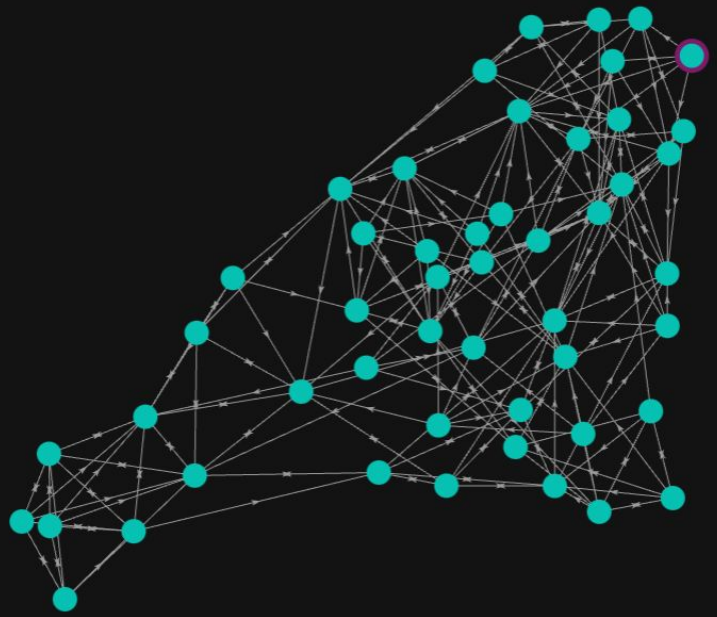
Visual Data Analysis with Vector Search

Studying your Data Visually

- MidJourney Dataset;
- Our **WebUI**;
- Our **Distance matrix API**

More details here:



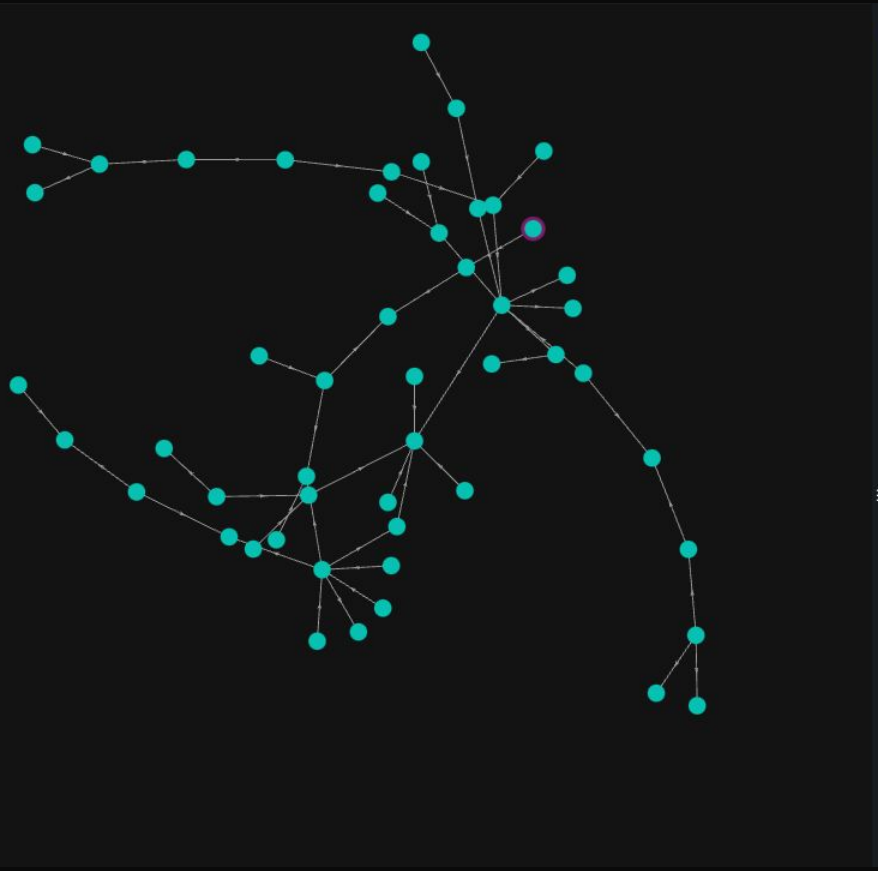


RUN

```
{
  "limit": 5,
  "sample": 50
}
```

...

id	999
file_name	662c52bf63c5da30405cfbcf_Eureka_Seven_V6_p.jpeg
image_url	https://storage.googleapis.com/demo-midjourney/images/662c52bf63c5da30405cfbcf_Eureka_Seven_V6_p.jpeg
name	Eureka Seven
url	/styles/eureka-seven

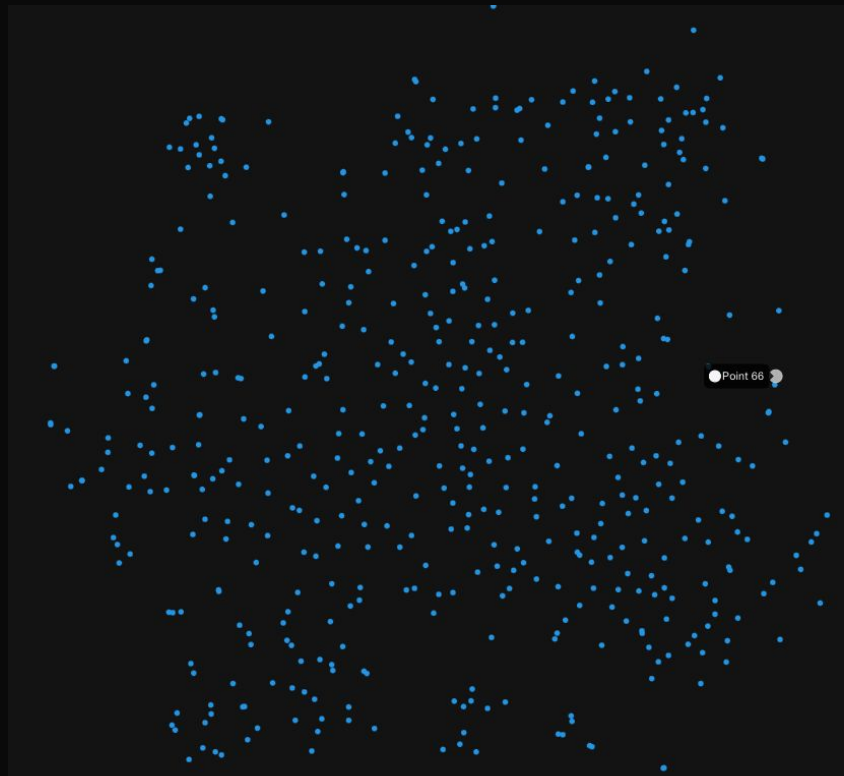


RUN

```
{
  "limit": 5,
  "sample": 50,
  "tree": true
}
```

...

id	4702
file_name	662a911d80e39cff0e63969c_Frank_Lloyd_Wright_V6_p.jpeg
image_url	https://storage.googleapis.com/demo-midjourney/images/662a911d80e39cff0e63969c_Frank_Lloyd_Wright_V6_p.jpeg
name	Frank Lloyd Wright
url	/styles/frank-lloyd-wright



RUN

```
{
  "limit": 500
}
```

id	66
file_name	662ba29aaa6720940d66a7bc_Serge_Attukwei_Clottey_V6_p.jpeg
image_url	https://storage.googleapis.com/demo-midjourney/images/662ba29aaa6720940d66a7bc_Serge_Attukwei_Clottey_V6_p.jpeg
name	Serge Attukwei Clottey
url	/styles/serge-attukwei-clottey

Vectors:

Vectors:

Default vector

Length: 512

OPEN GRAPH

How? Dimensionality Reduction

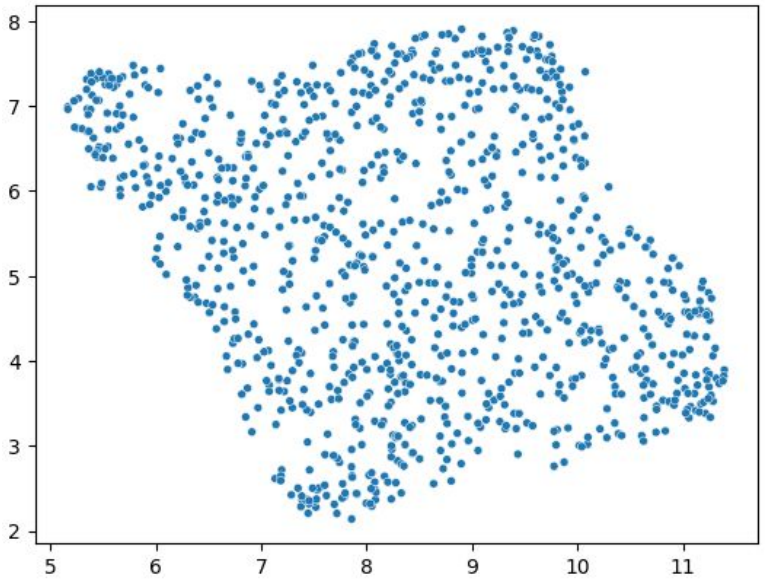
– to **visualize** high dimensional embeddings in 2D.

Dimensionality reduction algorithms:

- UMAP;
- t-SNE;
- Isomap;
- SpectralEmbedding…

All of them rely on building a **distance matrix**.

We can provide it directly. The "Visualization" tab in our WebUI makes use of this ability.



Clustering

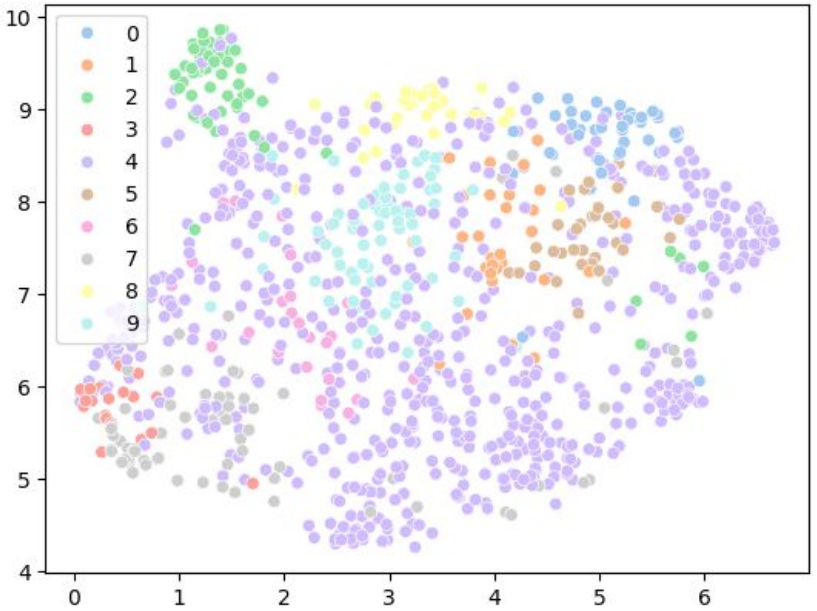
– to discover & define **data taxonomy.**

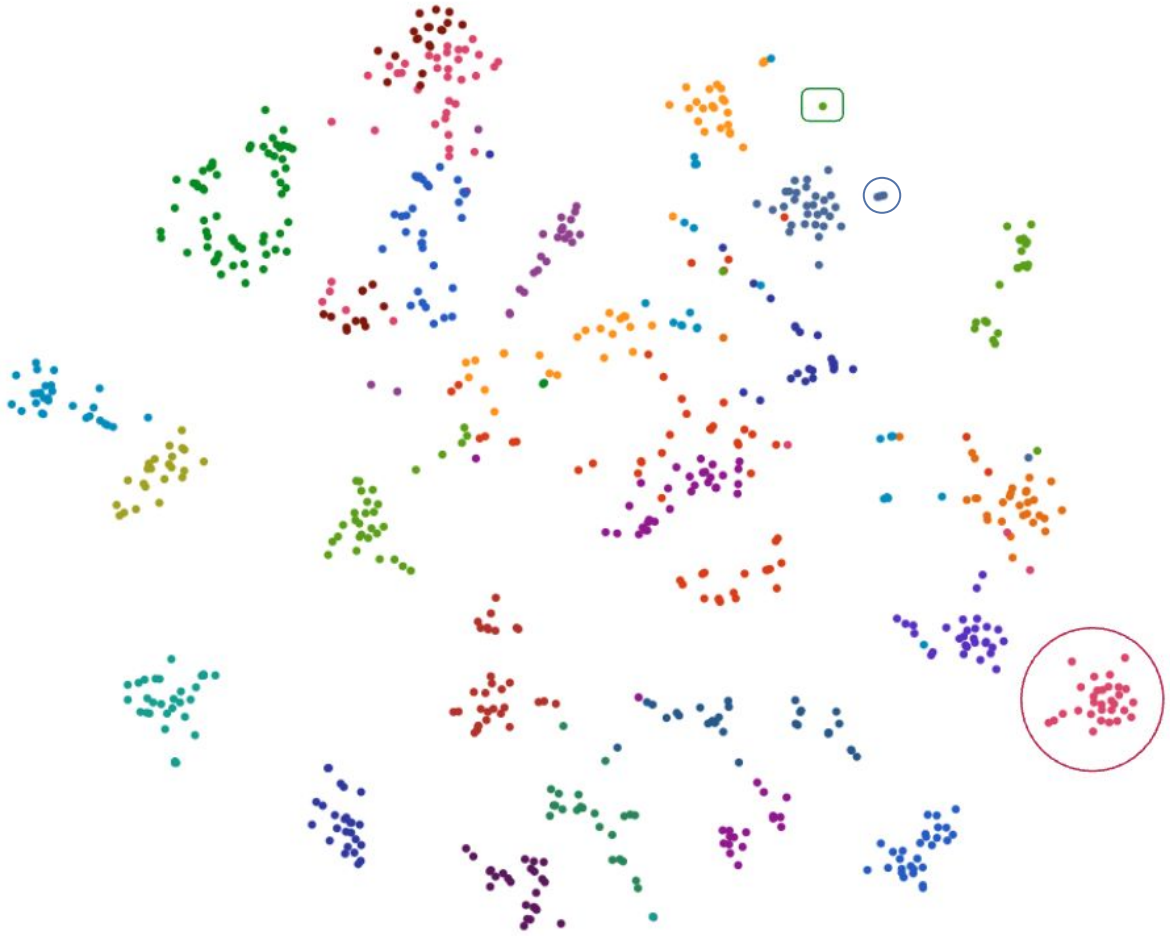
Clustering algorithms:

- K-means;
- DBSCAN;
- Agglomerative Clustering;
- Spectral Clustering…

All of them also rely on building a **distance matrix**.

Once again, **we can provide it directly**.



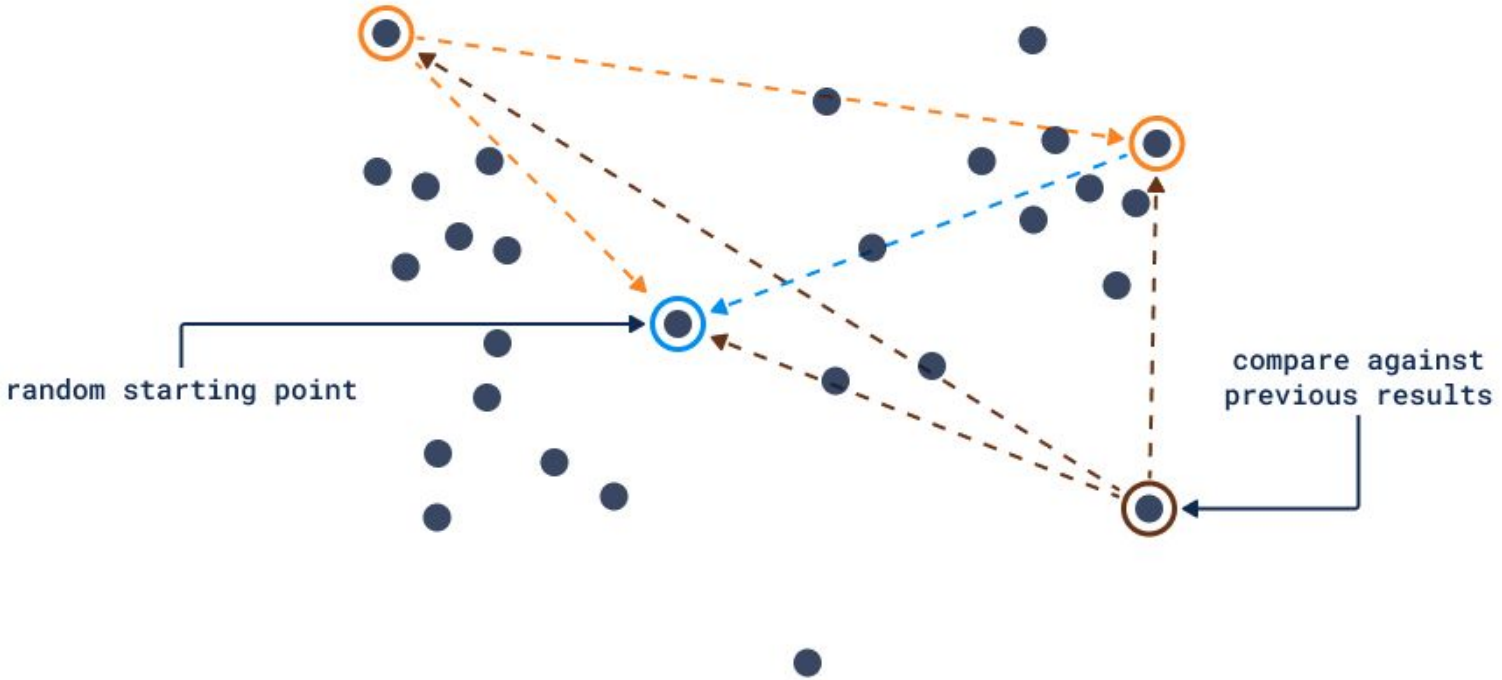


Vector Search for Scalable Data Analysis

How various is my data?

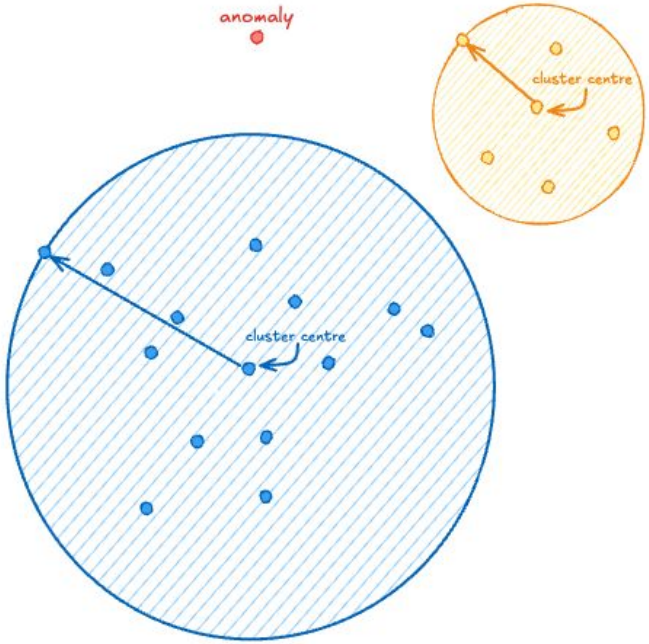
Random sampling might be biased toward more frequent types of documents.

Dissimilarity-based sampling incorporates previously selected vectors and iteratively returns dissimilar items.



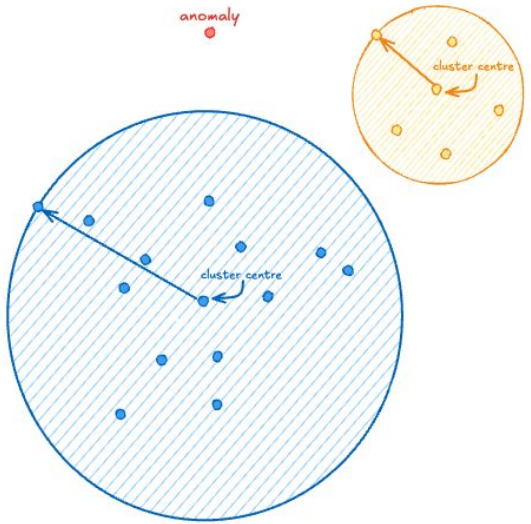
Dissimilarity-based (diversity) sampling as an iterative process.
In **Qdrant**, use the **Recommendation API** + **“best_score”** strategy, and pass all sampled points as **negative examples**.

Are there anomalies?



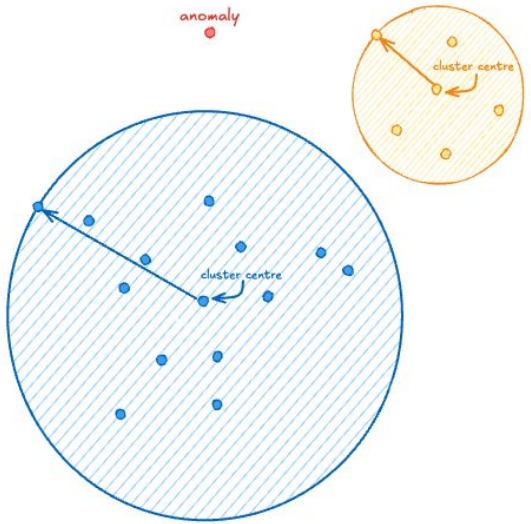
Are there anomalies?

1. Identify the **representative** of a cluster (class).
    a. medoid/centroid (distance matrix);
    b. category/class name;
    c. ideal candidate…



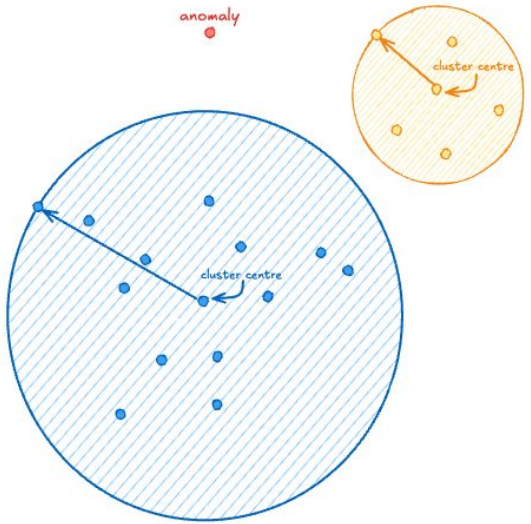
Are there anomalies?

1. Identify the **representative** of a cluster (class).
    a. medoid/centroid (distance matrix);
    b. category/class name;
    c. ideal candidate…
2. Calculate the **similarity score** between this representative and the **Xth furthest point** within the same class (X is meta parameter). Use **dissimilarity search**.



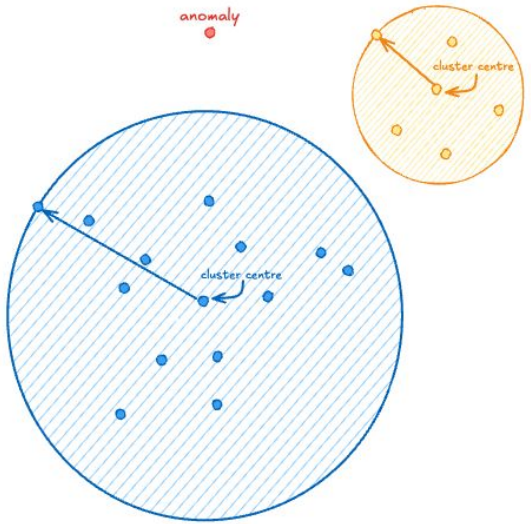
Are there anomalies?

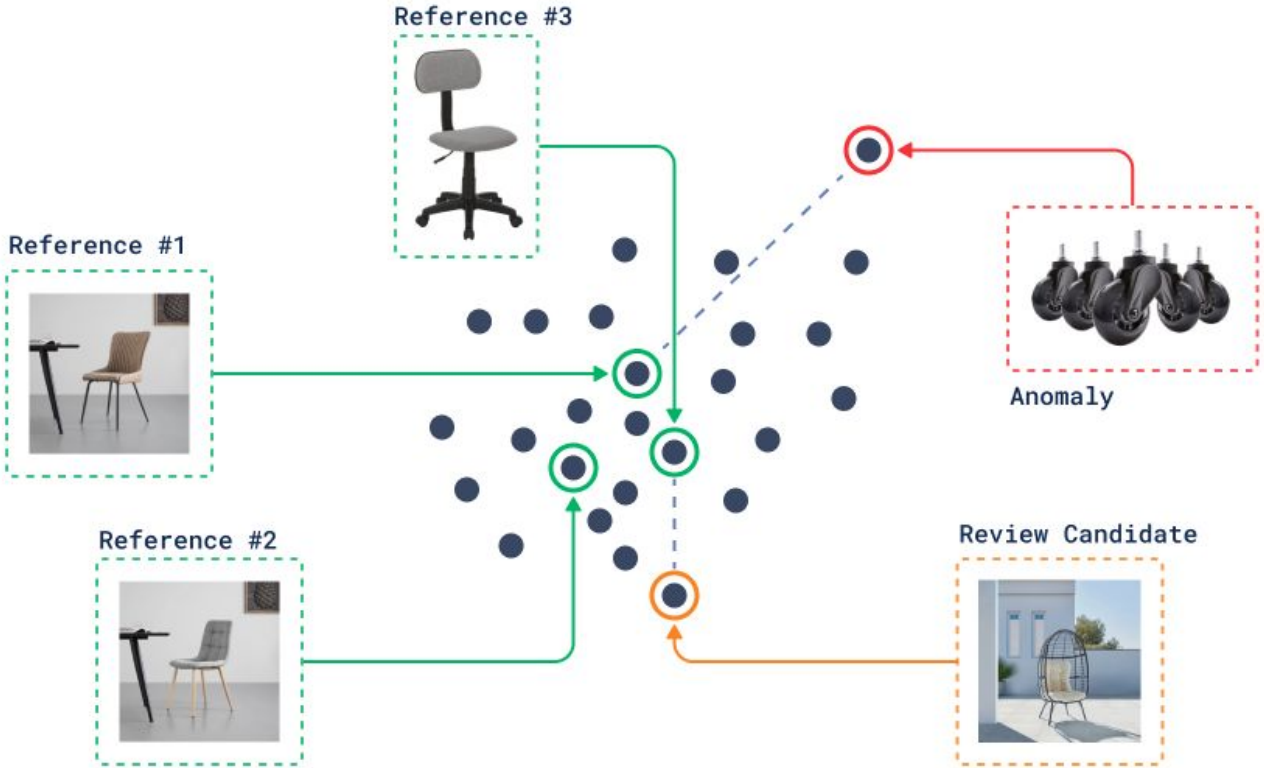
1. Identify the **representative** of a cluster (class).
    a. medoid/centroid (distance matrix);
    b. category/class name;
    c. ideal candidate…
2. Calculate the **similarity score** between this representative and the **Xth furthest point** within the same class (X is meta parameter). Use **dissimilarity search**.
3. Use this similarity score as the **threshold** (cluster border).



Are there anomalies?

1. Identify the **representative** of a cluster (class).
    a. medoid/centroid (distance matrix);
    b. category/class name;
    c. ideal candidate…
2. Calculate the **similarity score** between this representative and the **Xth furthest point** within the same class (X is meta parameter). Use **dissimilarity search**.
3. Use this similarity score as the **threshold** (cluster border).
4. Any data point with a similarity score lower than the threshold is considered an **anomaly for this cluster**.



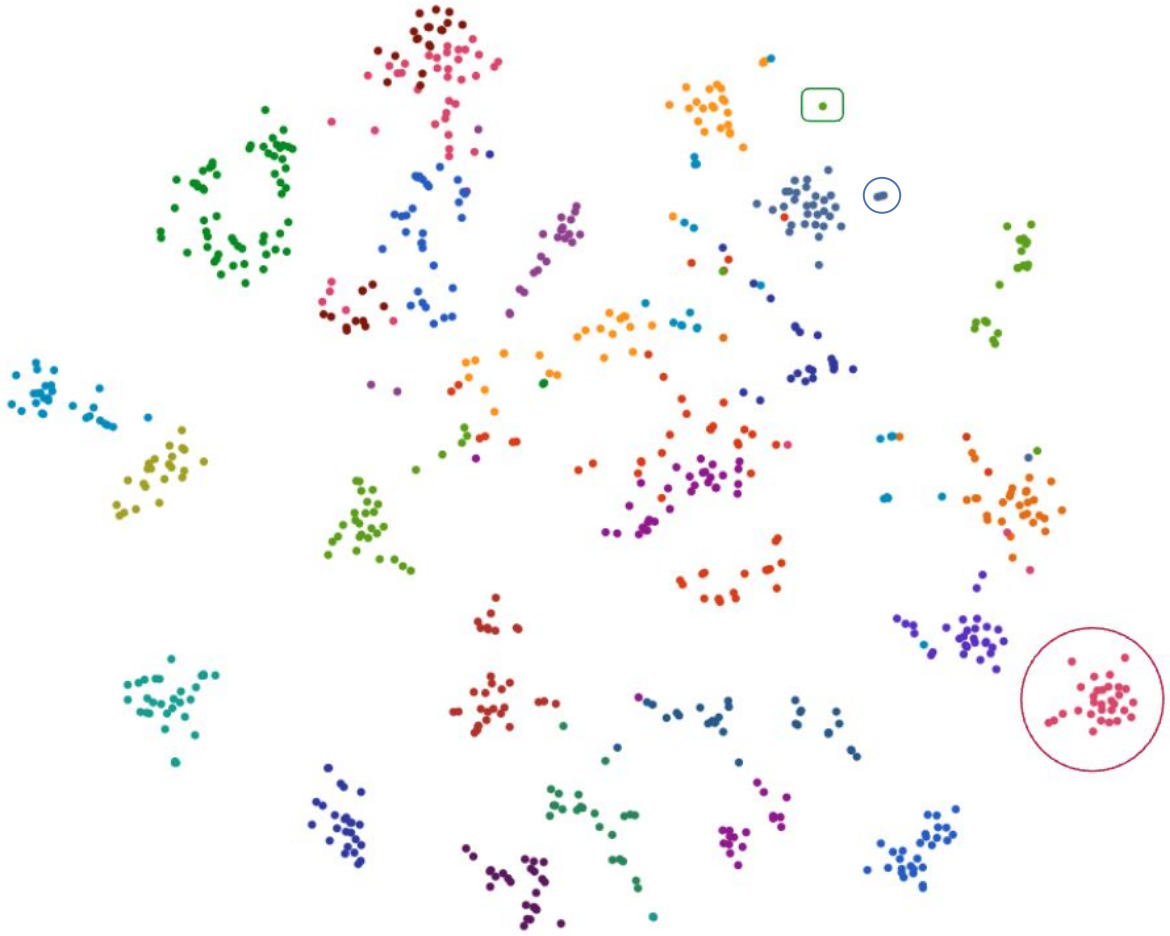


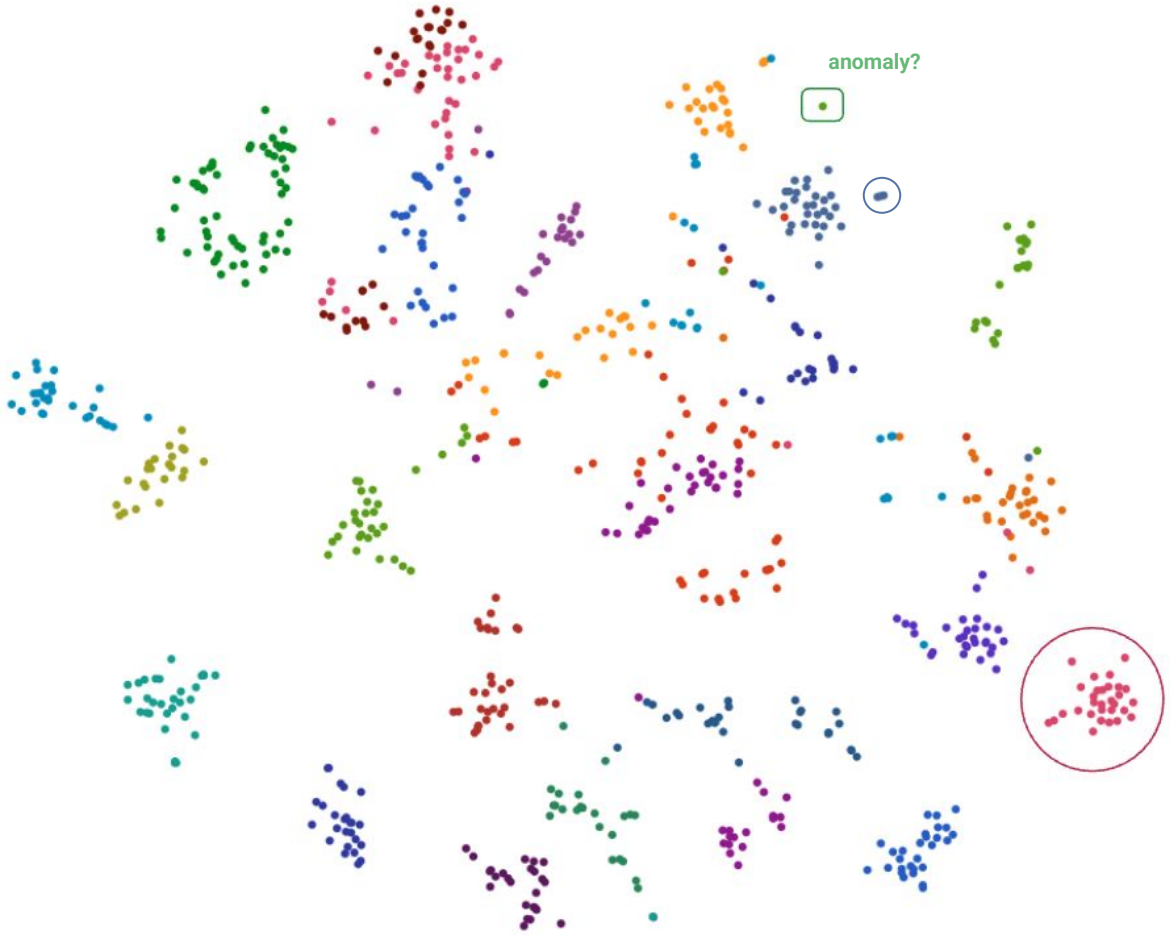
Outlier detection within a category

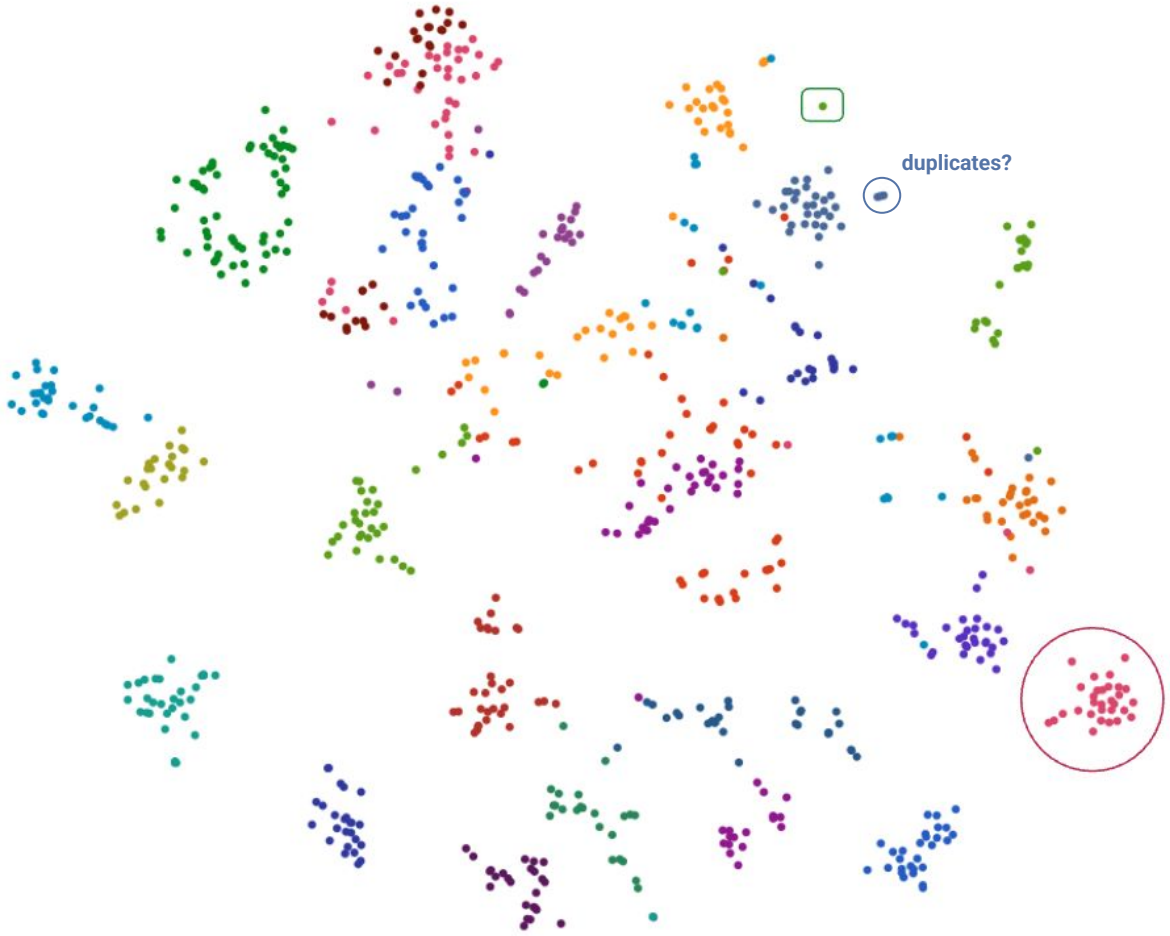
Mislabeling detection

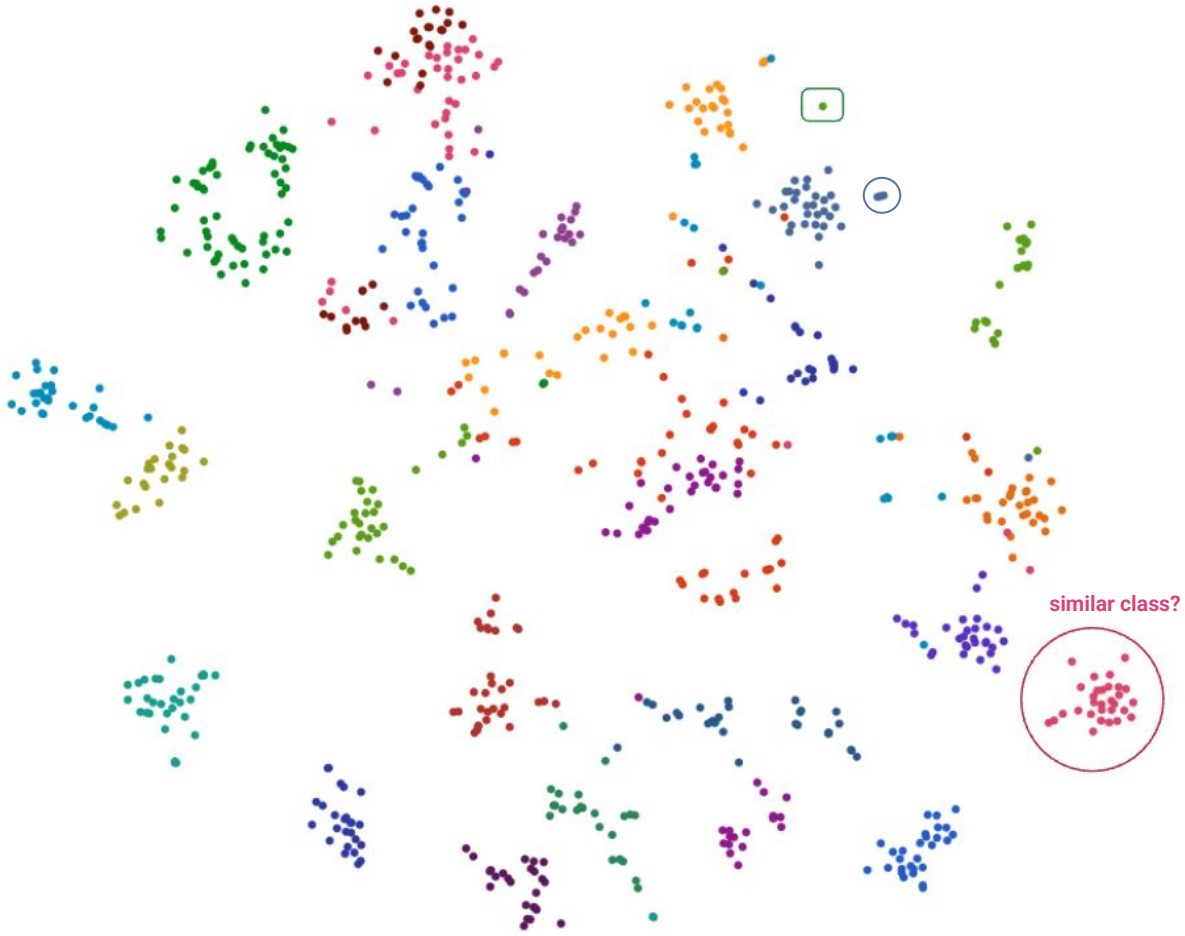
Query:
Chair

Vector Search-Based Data Analysis as a Business Solution




anomaly?


duplicates?


similar class?

Talking real use cases

- **Vertical farming**: Automating crop monitoring and detecting anomalies in plant growth.
- **Food industry**: Anomaly detection for quality control in food production.
- **Ecological monitoring**: Identifying islands of garbage or oil leaks in oceans.
- **Recruitment**: Matching CVs with suitable job positions through classification.
- **Spam detection**: Identifying and filtering out irrelevant or fraudulent content.
- **Content Cleanup:** Merging similar media or documentation around the same concepts.
- **Workflow Optimization:** Removing duplicates in codebases or tickets.

Happy to discuss more!

Evgeniya Sukhodolskaya
Developer Advocate
Qdrant

https://www.linkedin.com/in/evgeniya-sukhodolskaya/
https://x.com/krotenWanderung
https://qdrant.to/discord

drant

A free forever 1GB cluster included for trying out. No credit card required.