elastic

Search
a new era

David Pilato
@dadoonet
@pilato.fr

Elasticsearch

You Know, for Search

elastic

Elasticsearch

APACHE
LUCENE™

elastic

These are not the droids you are looking for.

```
GET /_analyze
{
  "char_filter": [ "html_strip" ],
  "tokenizer": "standard",
  "filter": [ "lowercase", "stop", "snowball" ],
  "text": "These are <em>not</em> the droids
           you are looking for."
}
```

```
"char_filter": "html_strip"
```

```
These are <em>not</em> the droids you are
   looking for.
```

```
These are not the droids you are
   looking for.
```

```
"tokenizer": "standard"
```

These are not the droids you are looking for.

These
are
not
the
droids
you
are
looking
for

```
"filter": "lowercase"
```

These	these
are	are
not	not
the	the
droids	droids
you	you
are	are
looking	looking
for	for

```
"filter": "stop"
```

These	these	
are	are	
not	not	
the	the	
droids	droids	droids
you	you	you
are	are	
looking	looking	looking
for	for	

elastic

```
"filter": "snowball"
```

These		these				
are		are				
not		not				
the		the				
droids	→	droids	→	droids	→	droid
you		you		you		you
are		are				
looking		looking		looking		look
for		for				

elastic

These are not the **droids** **you** are **looking** for.

```
{ "tokens": [{
      "token": "droid",
      "start_offset": 27, "end_offset": 33,
      "type": "<ALPHANUM>", "position": 4
    },{
      "token": "you",
      "start_offset": 34, "end_offset": 37,
      "type": "<ALPHANUM>", "position": 5
    }, {
      "token": "look",
      "start_offset": 42, "end_offset": 49,
      "type": "<ALPHANUM>", "position": 7
    }]}
```

elastic

Semantic
search
≠
Literal
matches

Elasticsearch

You Know, for Search

elastic

Elasticsearch

You Know, for Vector Search

elastic

What is a Vector?

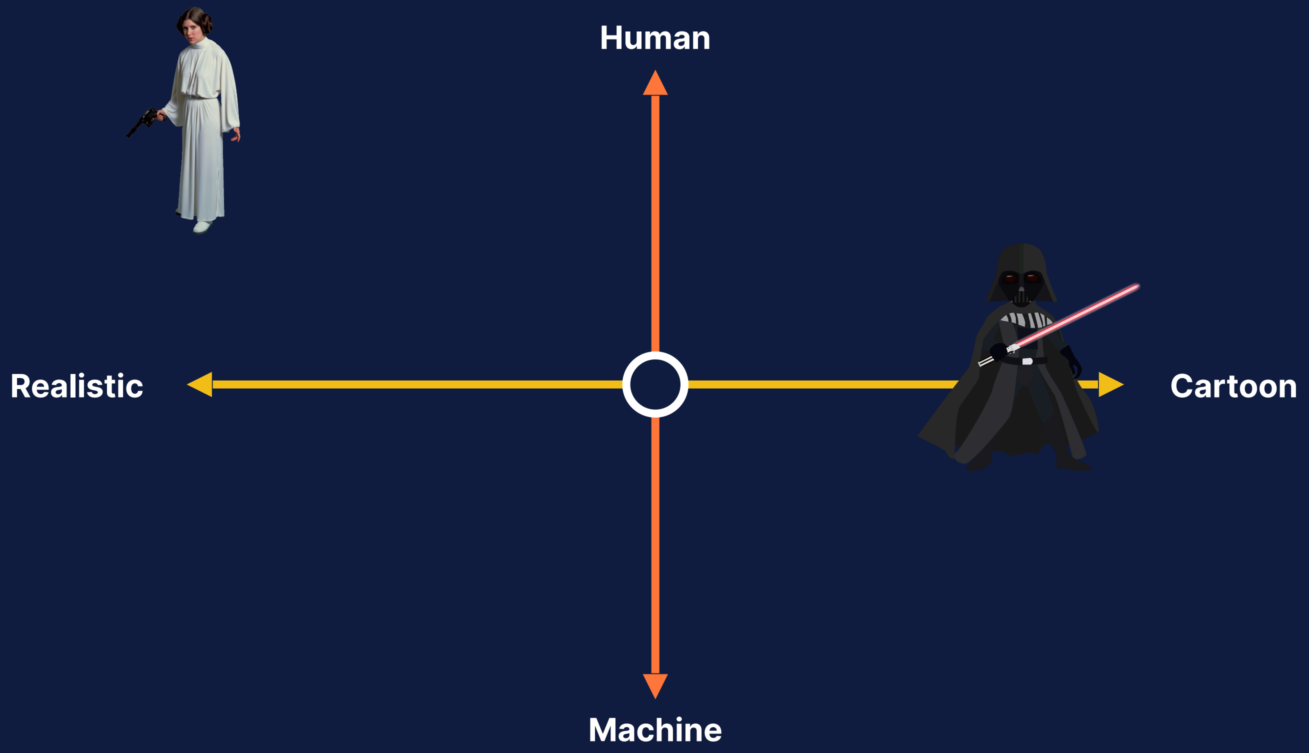
elastic

Embeddings represent your data

Example: 1-dimensional vector

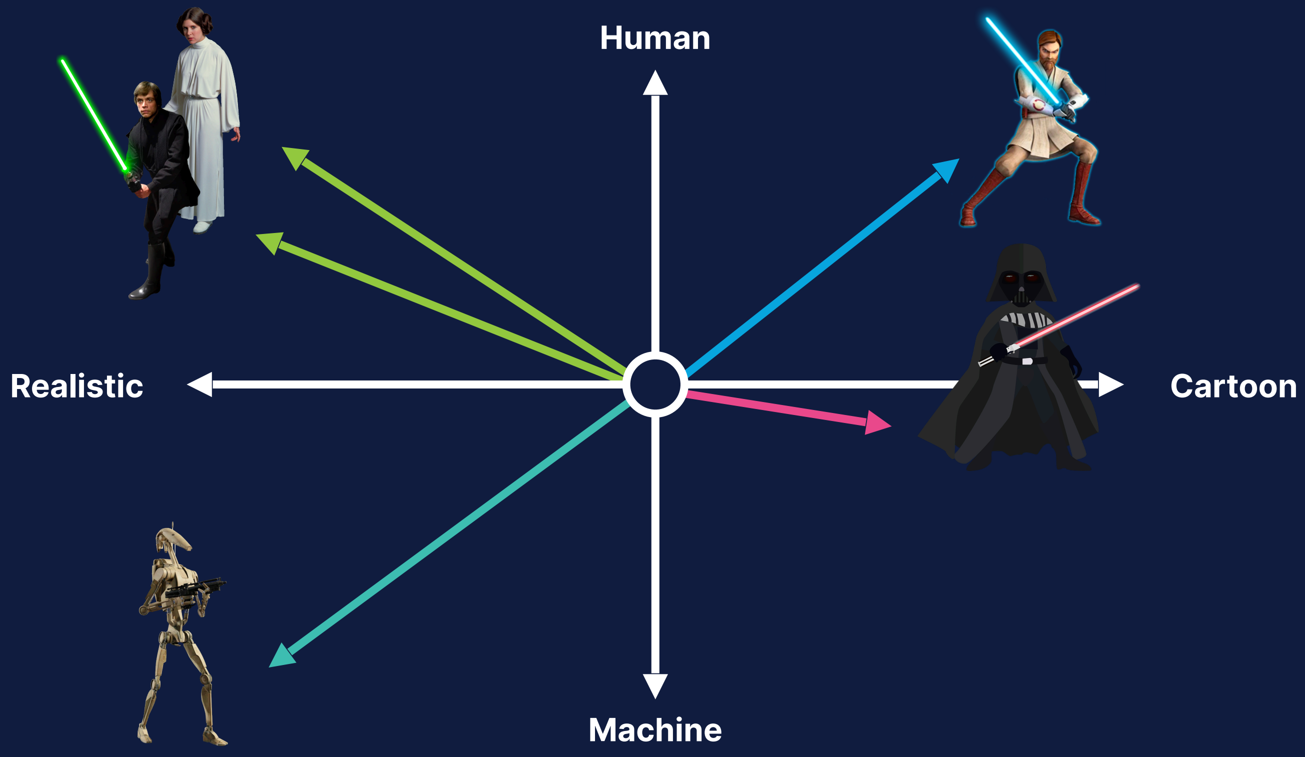
Character	Vector
	[-1]
	[1]

Multiple dimensions
represent different data aspects



Character	Vector
	[-1, 1]
	[1, 0]

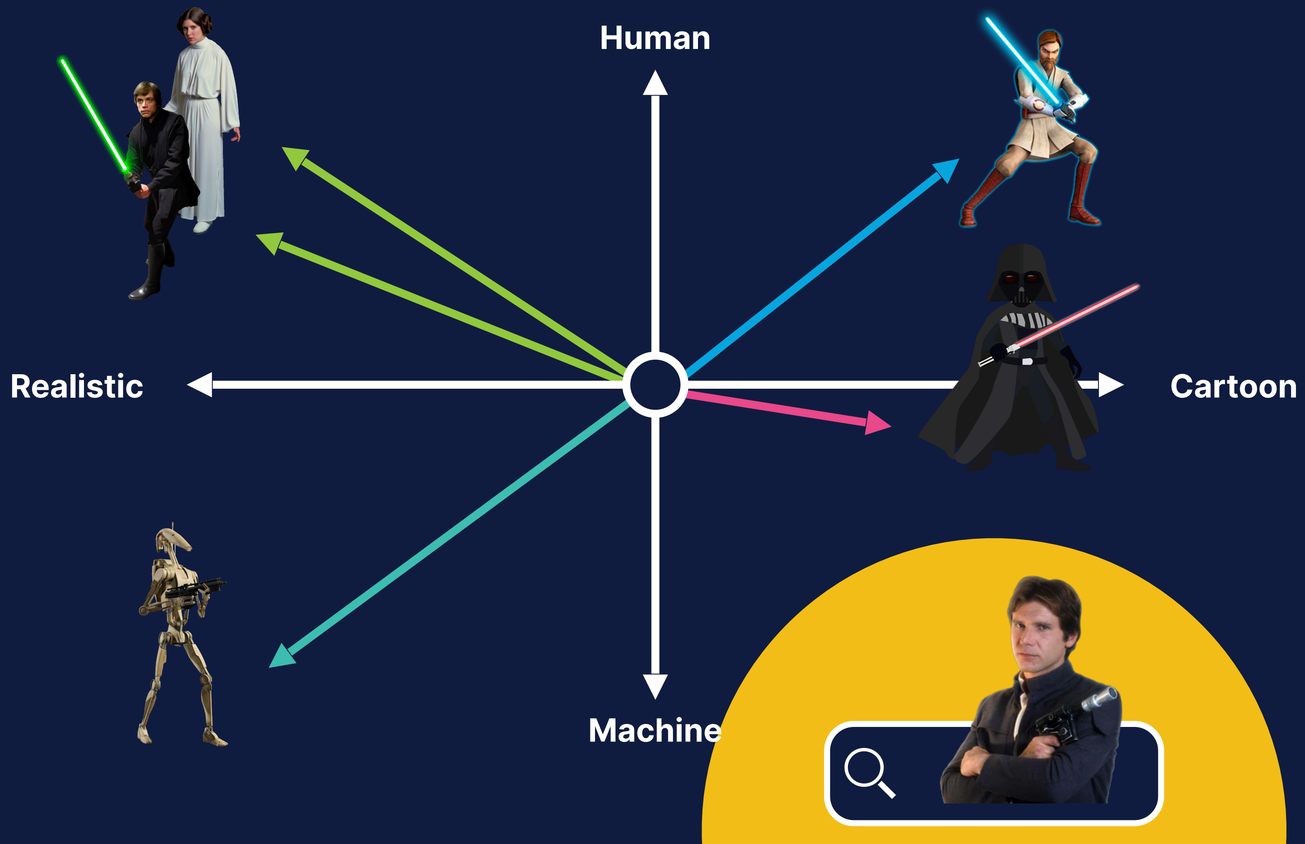
Similar data
is grouped together



Character	Vector
	[-1.0, 1.0]
	[1.0, 0.0]
	[-1.0, 0.8]
	[1.0, 1.0]
	[-1.0, -1.0]

elastic

Vector search ranks objects
by similarity (~relevance) to the query



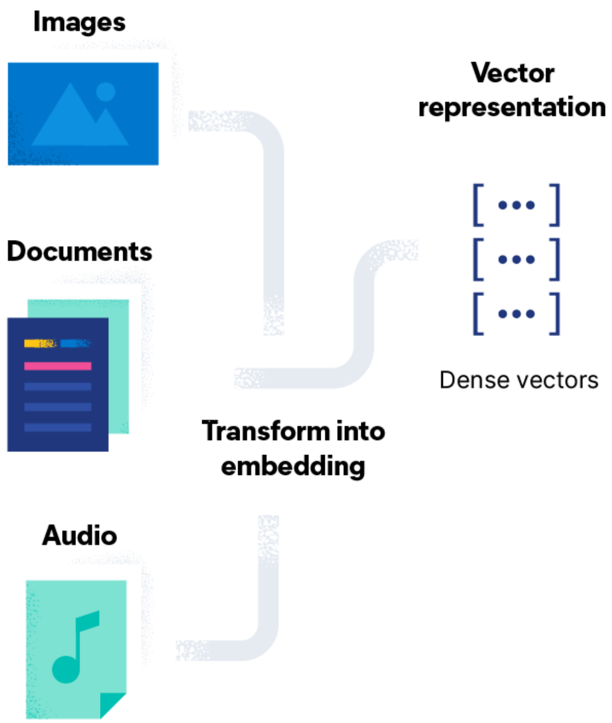
Rank	Result
Query	
1	
2	
3	
4	
5	

elastic

How do you

index vectors?

Architecture of Vector Search



Choice of Embedding Model

Start with Off-the Shelf Models

- Text data: Hugging Face (like Microsoft's E5)
- Images: OpenAI's CLIP

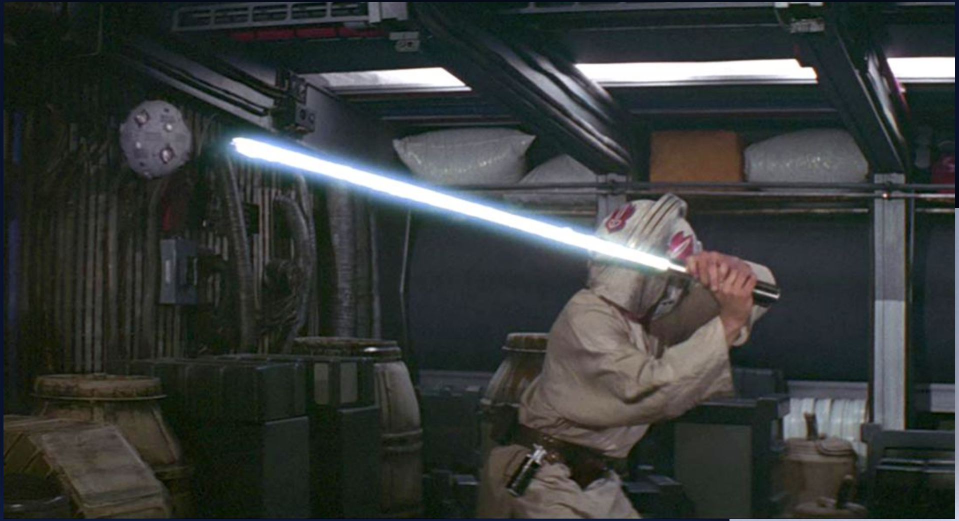
Extend to Higher Relevance

- Apply hybrid scoring
- Bring Your Own Model: requires expertise + labeled data

elastic

Problem

training vs actual use-case



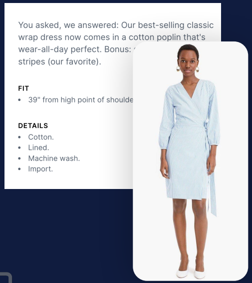
elastic

dense_vector field type

```
PUT ecommerce
{
  "mappings": {
    "properties": {
      "description": {
        "type": "text"
      }
      "desc_embedding": {
        "type": "dense_vector"
      }
    }
  }
}
```

elastic

Data Ingestion and Embedding Generation



Source data

POST /ecommerce/_doc

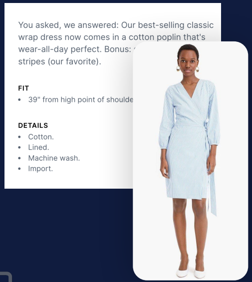
```
{
  "_id":"product-1234",
  "product_name":"Summer Dress",
  "description":"Our best-selling…",
  "Price": 118,
  "color":"blue",
  "fabric":"cotton",
} "desc_embedding":[0.452,0.3242,…],
} "img_embedding":[0.012,0.0,…]
}
```

POST /ecommerce/_doc

elastic

With Elastic ML

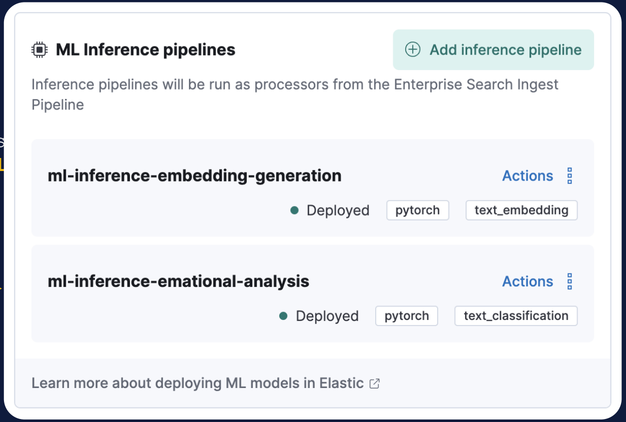
commercial



Source data

```
{
  "_id":"product-1234",
  "product_name":"Summer Dres
  "description":"Our best-sel
  "Price": 118,
  "color":"blue",
  "fabric":"cotton",
}
```

POST /ecommerce/_doc




```
{
  "_id":"product-1234",
  "product_name":"Summer Dress",
  "description":"Our best-selling…",
  "Price": 118,
  "color":"blue",
  "fabric":"cotton",
  "desc_embedding":[0.452,0.3242,…]
}
```

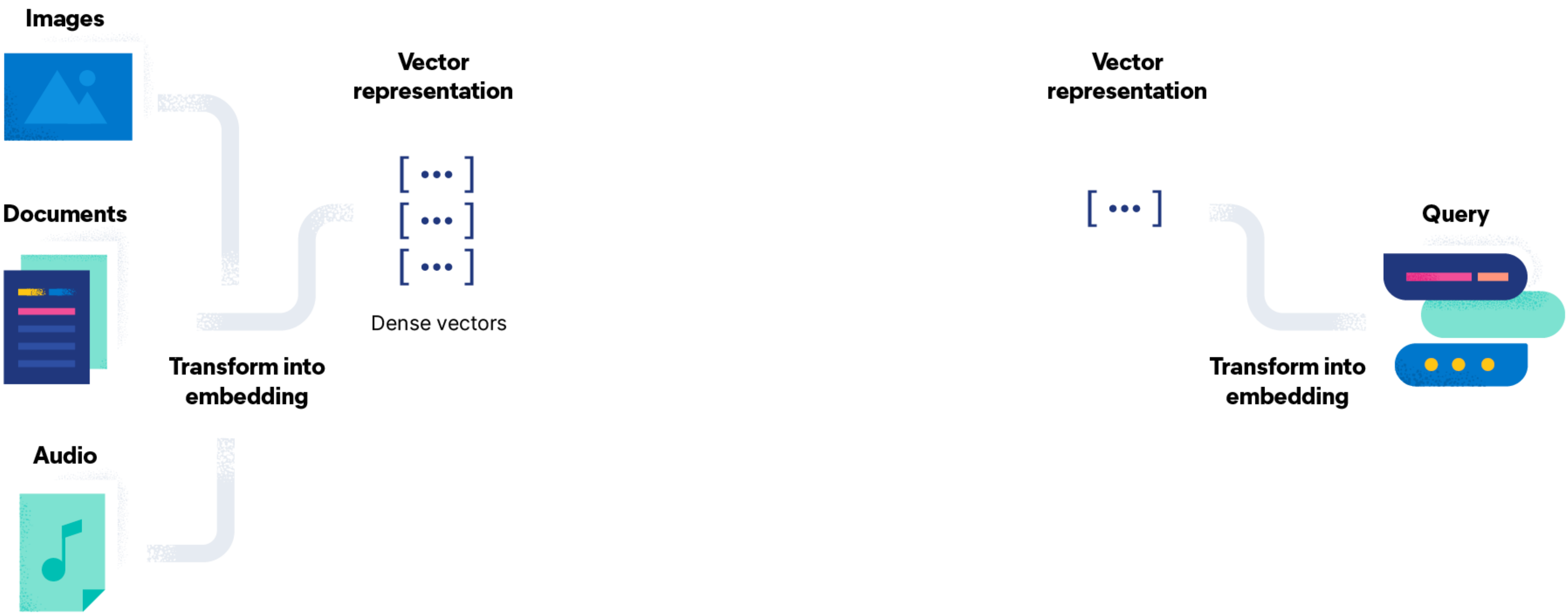
elastic

How do you

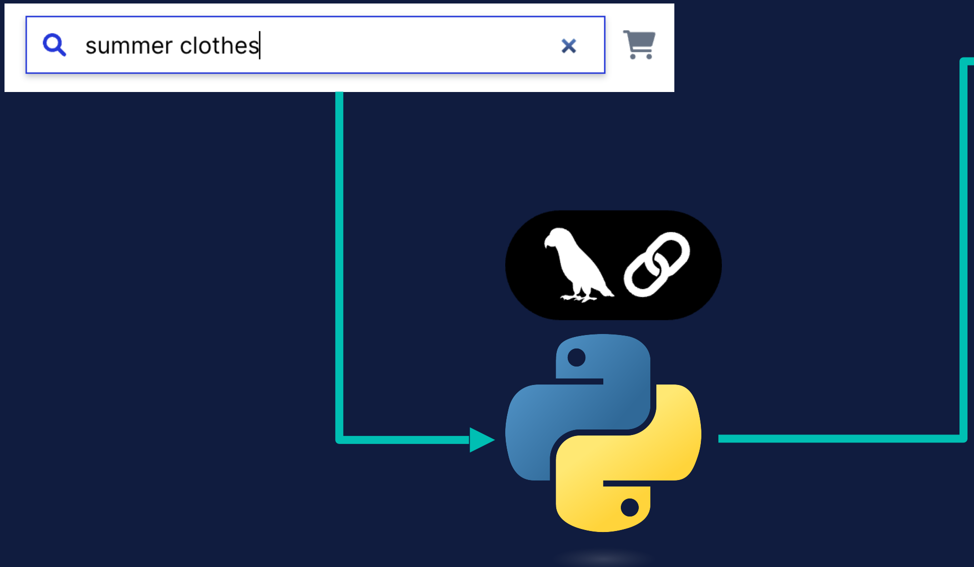
search **vectors**?

elastic

Architecture of Vector Search



knn query




```
GET /ecommerce/_search
{
 "query" : {
    "bool": {
      "must": [{
         "knn": {
           "field": "desc_embbeding",
           "query_vector": [0.123, 0.244,...]



         }
      }],
      "filter": {
        "term": {
          "department": "women"
        }
      }
    }
  },
  "size": 10
}
```

commercial

knn query (with Elastic ML)


```
GET /ecommerce/_search
{
 "query" : {
    "bool": {
      "must": [{
         "knn": {
           "field": "desc_embbeding",
           "query_vector_builder": {
             "text_embedding": {
              "model_text": "summer clothes",
              "model_id": <text-embedding-model>
             }
           }
         }
      }],
      "filter": {
        "term": {
          "department": "women"
        }
      }
    }
  },
  "size": 10
}
```

Transformer model

elastic

semantic_text field type

new from 8.15

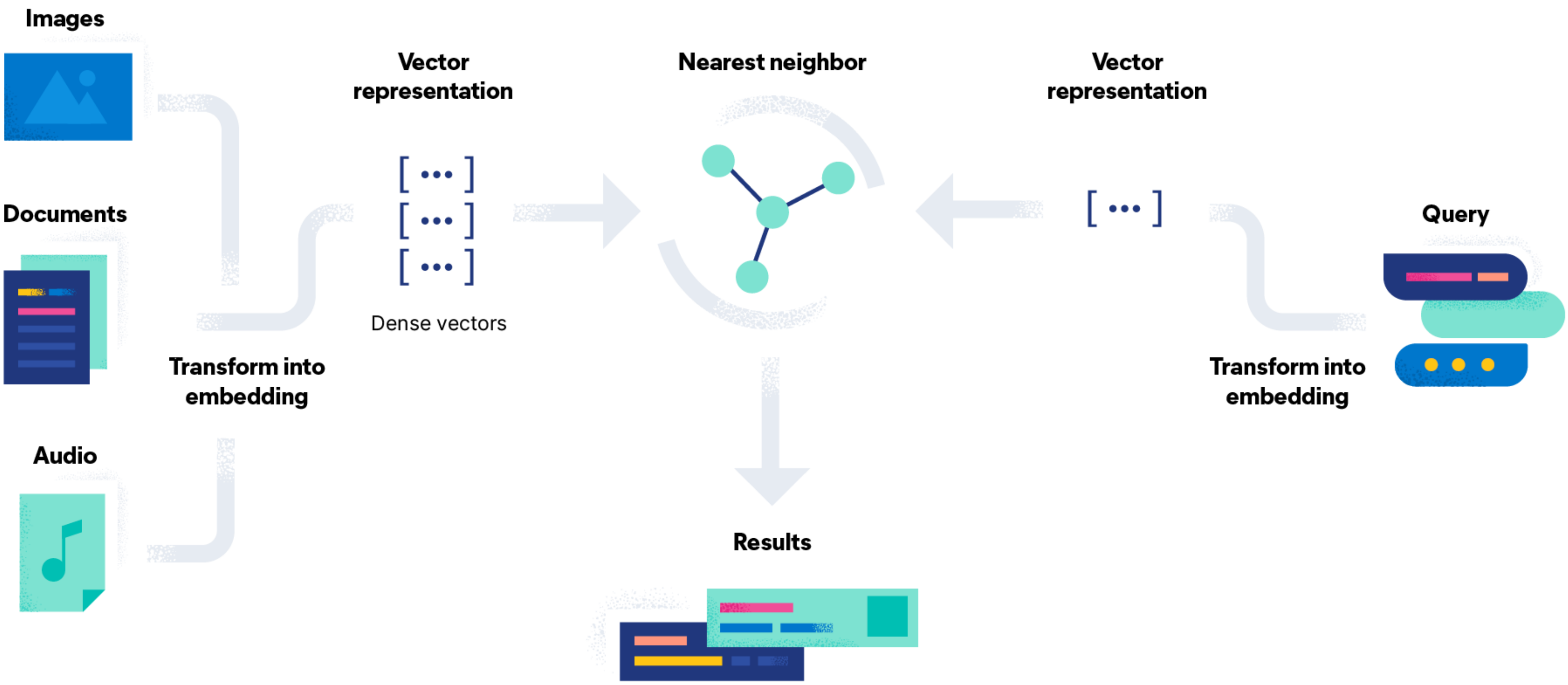
```
PUT ecommerce
{
  "mappings": {
    "properties": {
      "description": {
        "type": "text",
        "copy_to": [ "desc_embedding" ]
      }
      "desc_embedding": {
        "type": "semantic_text"
      }
    }
  }
}
```

```
POST ecommerce/_doc
{
  "description": "Our best-selling…"
}
```

```
GET ecommerce/_search
{
  "query": {
    "semantic": {
      "field": "desc_embedding"
      "query" : "I'm looking for a red dress for a DJ party"
}}}
```

elastic

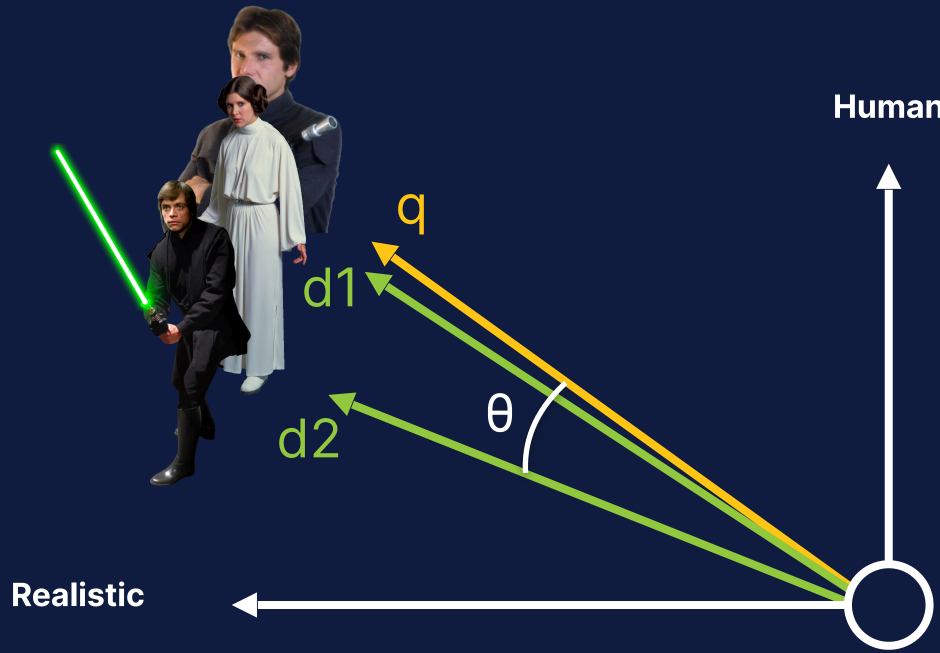
Architecture of Vector Search



But how does it

really work?

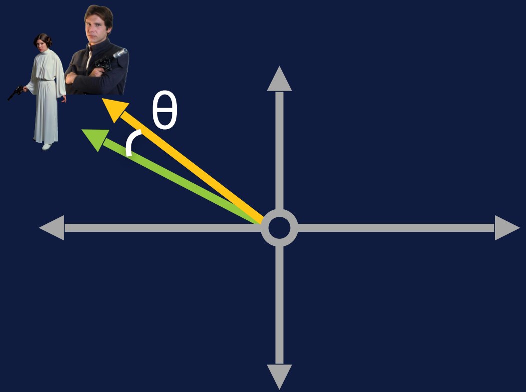
Similarity



$$cos(\theta) = \frac{\vec{q} \times \vec{d}}{|\vec{q}| \times |\vec{d}|}$$

$$\_score = \frac{1 + cos(\theta)}{2}$$

Similarity: cosine (cosine)

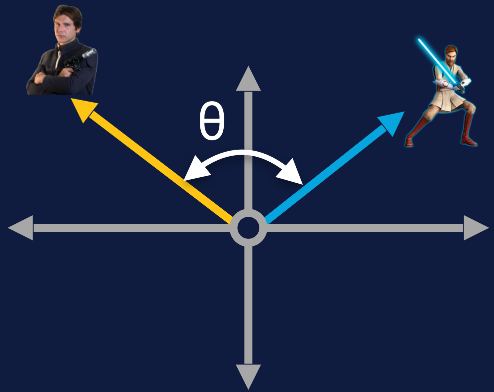


Similar vectors

θ close to 0

cos(θ) close to **1**

$$\_score = \frac{1+1}{2} = 1$$

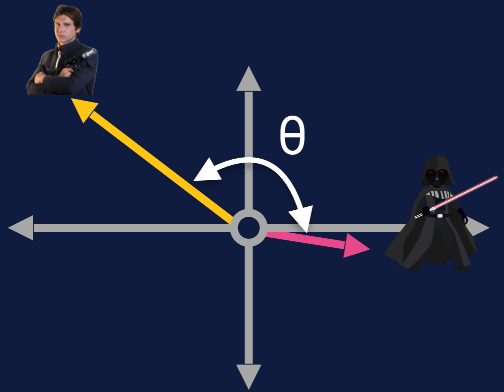


Orthogonal vectors

θ close to 90°

cos(θ) close to **0**

$$\_score = \frac{1+0}{2} = 0.5$$



Opposite vectors

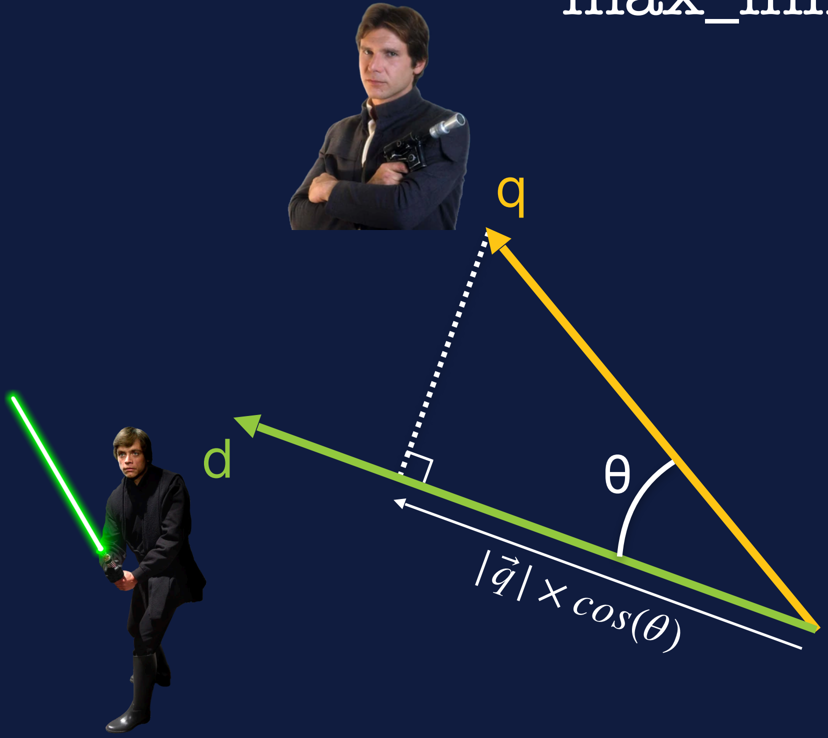
θ close to 180°

cos(θ) close to **-1**

$$\_score = \frac{1-1}{2} = 0$$

elastic

Similarity: Dot Product (dot_product or max_inner_product)



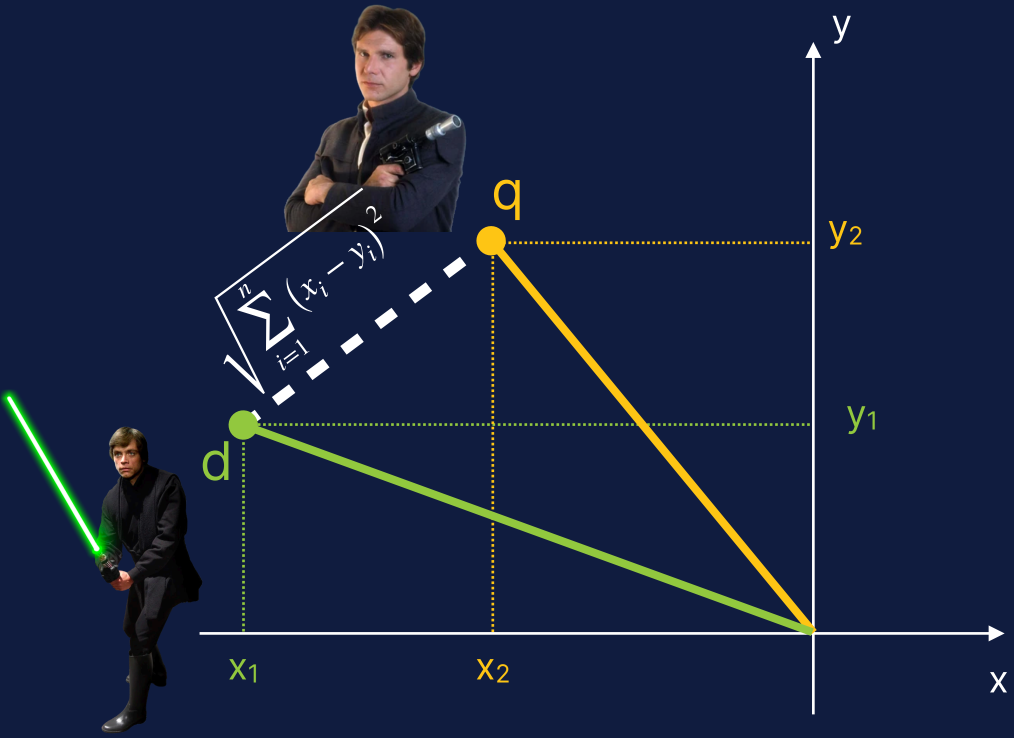
$$\vec{q} \times \vec{d} = |\vec{q}| \times cos(\theta) \times |\vec{d}|$$

$$\_score_{float} = \frac{1 + dot\_product(q, d)}{2}$$

$$\_score_{byte} = \frac{0.5 + dot\_product(q, d)}{32768 \times dims}$$

elastic

Similarity: Euclidean distance (l2_norm)



$$l2\_norm_{q,d} = \sqrt{\sum_{i=1}^{n} \left(x_i - y_i\right)^2}$$

$$\_score = \frac{1}{1 + (l2\_norm_{q,d})^2}$$

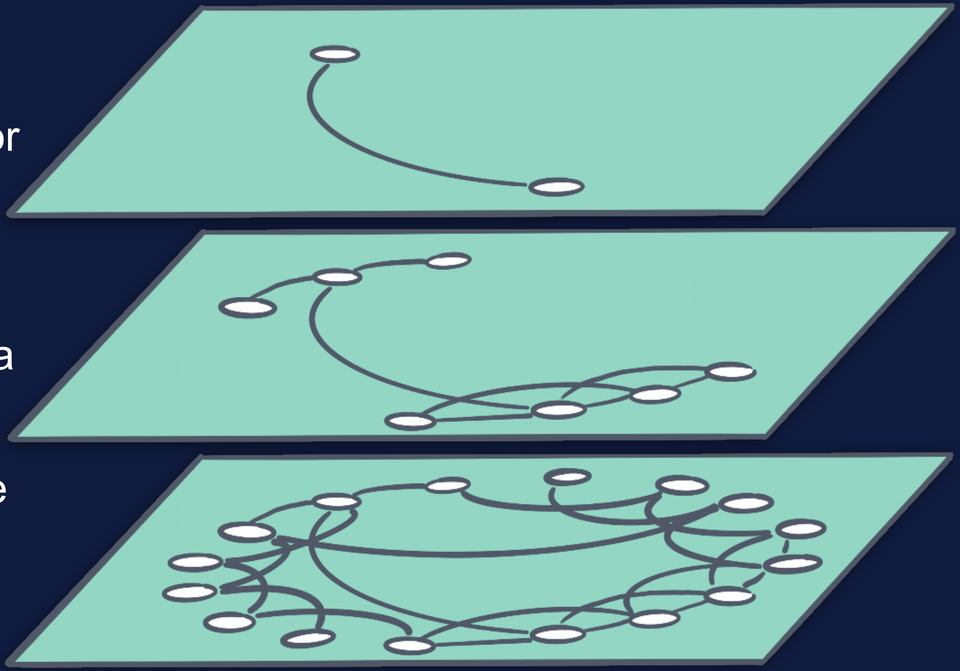
Brute Force

elastic

Hierarchical Navigable Small Worlds (HNSW)

One popular approach

- **HNSW:** a layered approach that simplifies access to the nearest neighbor
- **Tiered:** from coarse to fine approximation over a few steps
- **Balance:** Bartering a little accuracy for a lot of scalability
- **Speed:** Excellent query latency on large scale indices



elastic

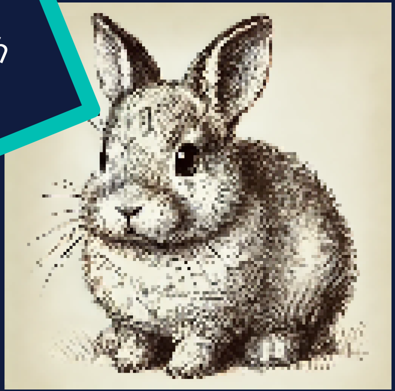
Scalar Quantization

Elasticsearch 8.14+ default

float32

Recall: High
Precision: High
Rescore: Likely Not Needed

Full RAM Required



int8

Recall: Good
Precision: Good
Oversampling: Moderate

Rescore: Reasonable

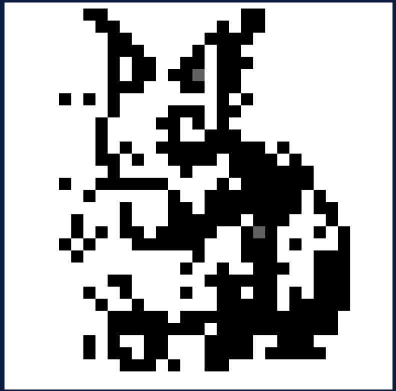
4X RAM Savings

int4

Recall: Low
Precision: Low
Oversampling: Needed

Rescore: may be slower

8X RAM Savings



bit

Recall: Bad
Precision: Bad
Oversampling: Needed

Rescore: Expensive and Limiting

32X RAM Savings

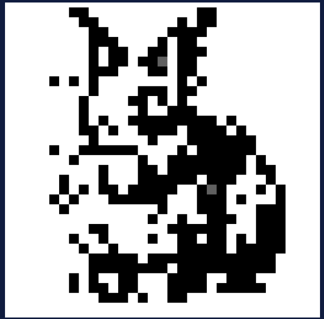
elastic

BBQ aka Better Binary Quantization

float32

int8

int4

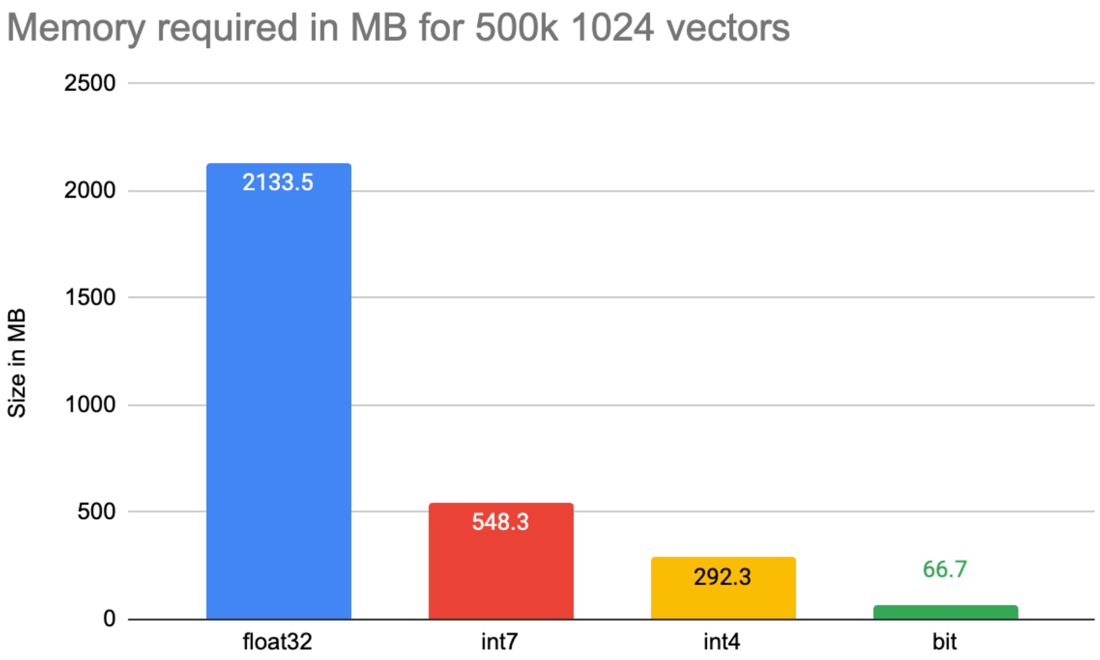


bit

BBQ*

BBQ: 32X RAM savings.
Faster & more accurate than Product Quantization

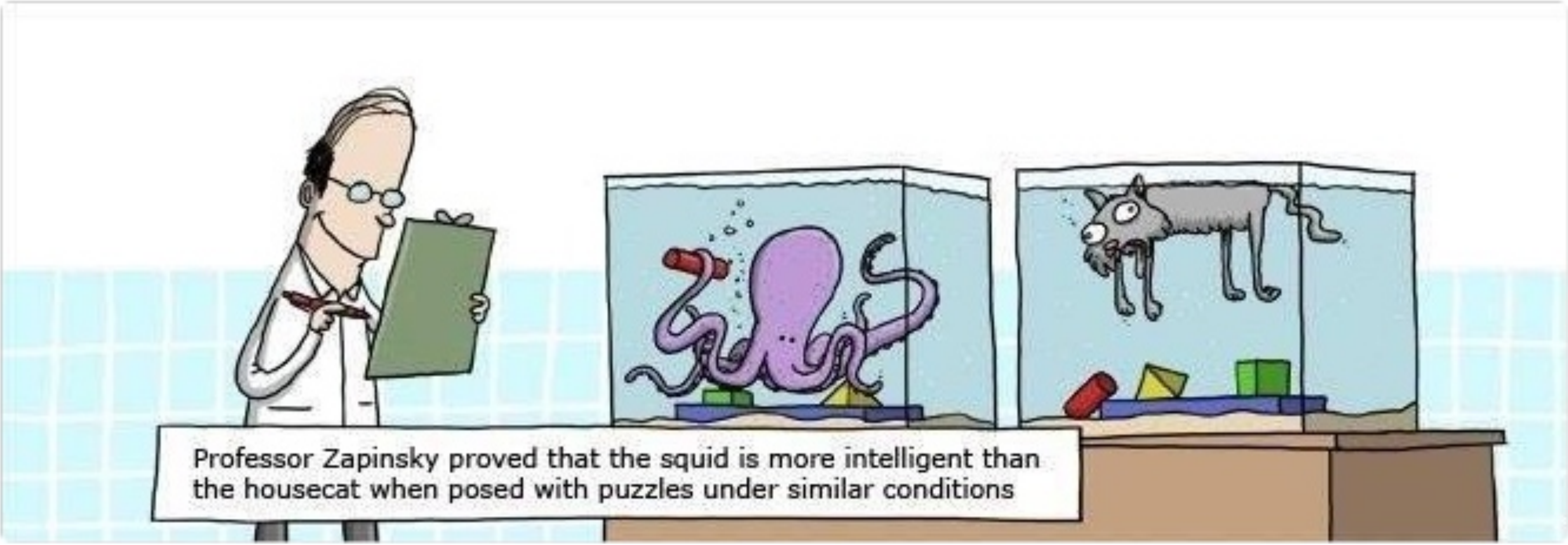
Memory required



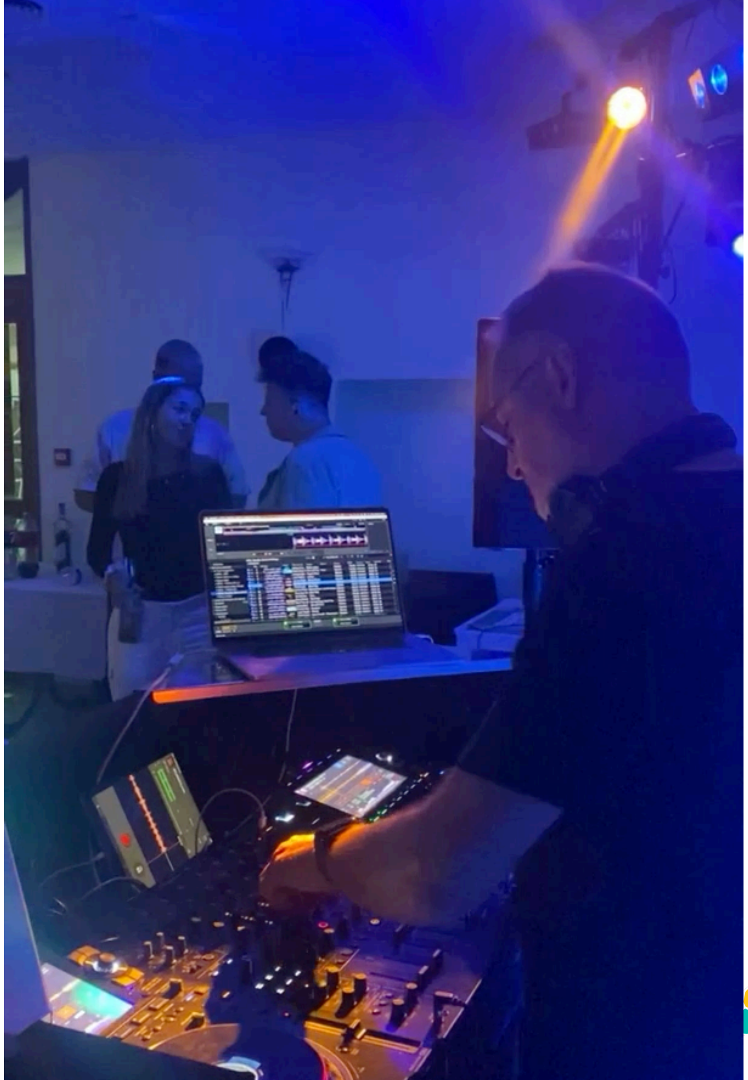
100M vectors?
Only 12GB!?! One single node.

elastic

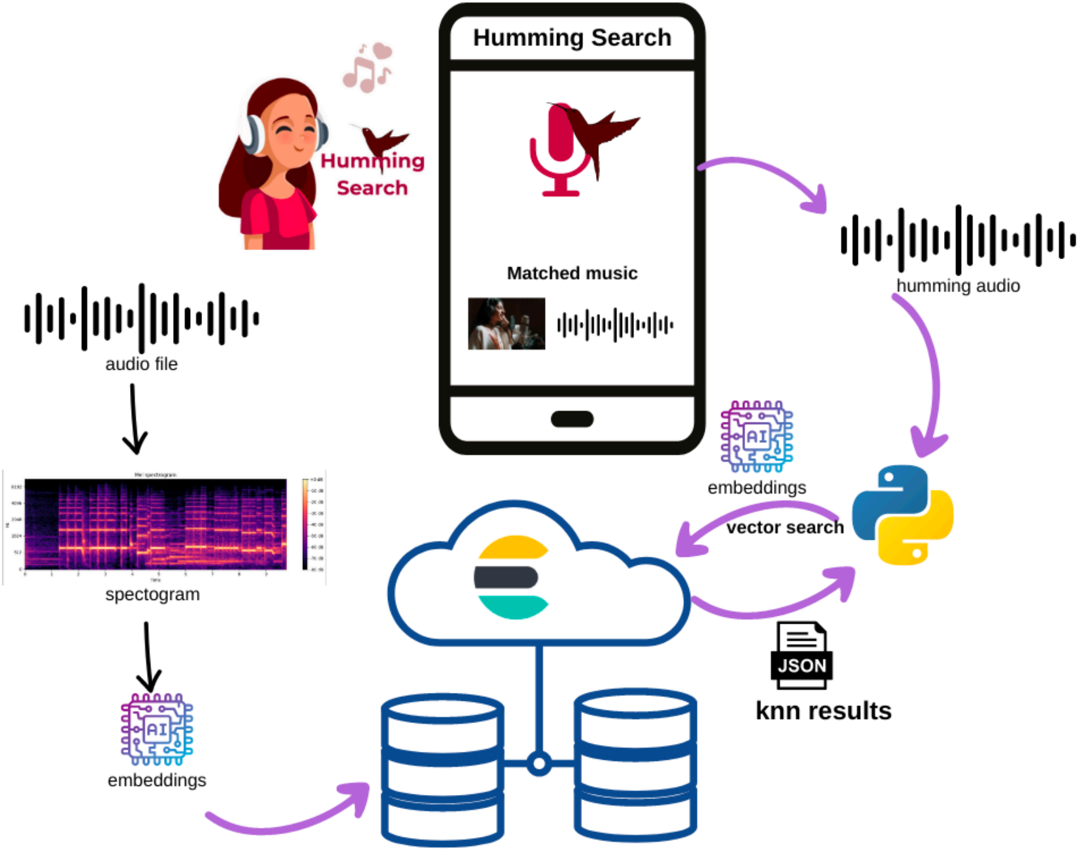
Benchmarketing



https://djdadoo.pilato.fr/



elastic



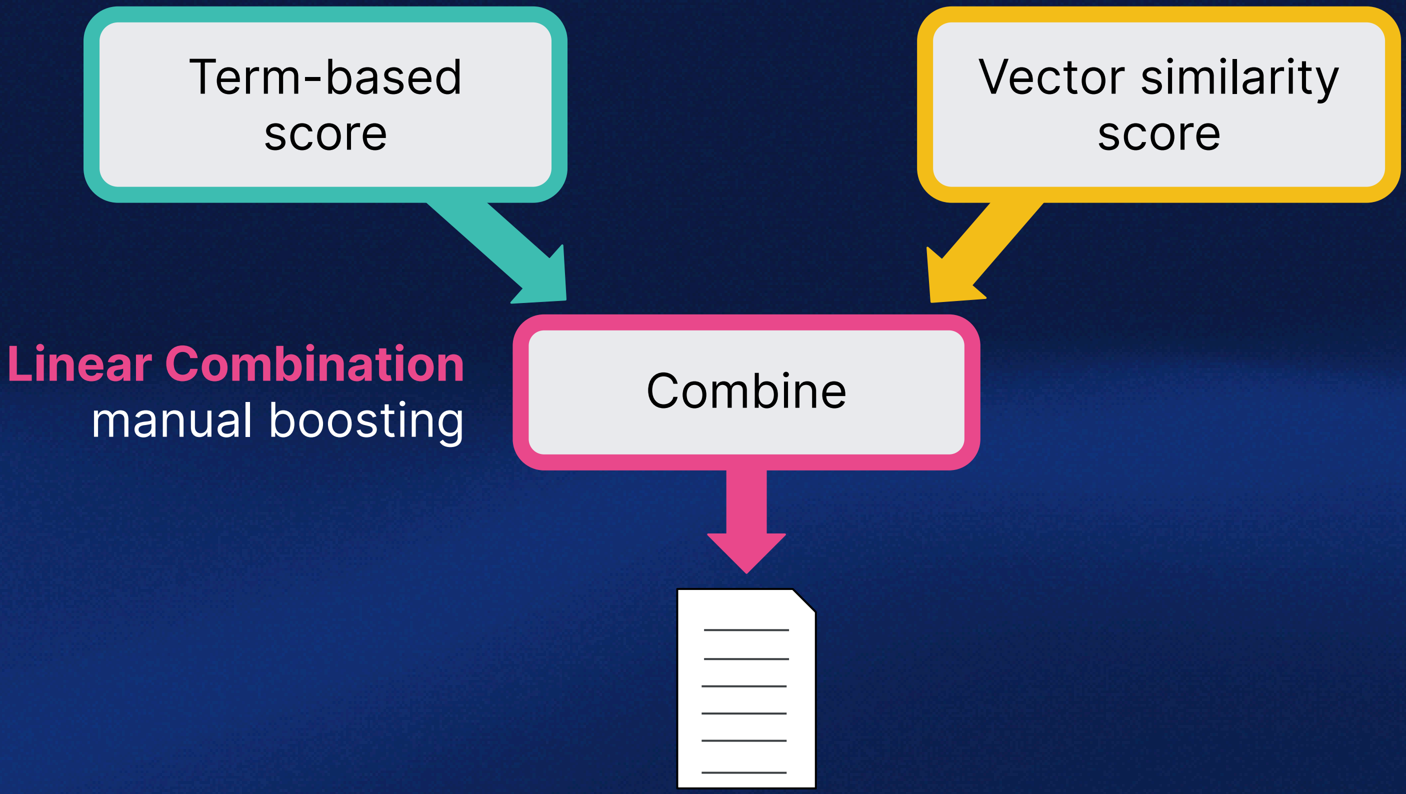
https://github.com/dadoonet/music-search/

Elasticsearch

You Know, for **Hybrid** Search

elastic

Hybrid scoring



elastic

Manual boosting

```
GET ecommerce/_search
{
  "query" : {
    "bool" : {
      "must" : [{
        "match": {
          "description": {
            "query": "summer clothes"
          }
        }
      },{
        "semantic": {
          "field": "desc_embbeding",
          "query": "summer clothes",
          "boost": 100.0
        }
      }]
    }
  }
}
```

```
PUT starwars
{
  "mappings": {
    "properties": {
      "text.tokens": {
        "type": "sparse_vector"
      }
    }
  }
}
```

"These are not the droids you are looking for.",

"Obi-Wan never told you what happened to your father."

```
GET starwars/_search
{
   "query":{
      "sparse_vector": {
        "field": "text.tokens",
        "query_vector": { "lucas": 0.50047517,
                          "ship": 0.29860738,
                          "dragon": 0.5300422,
                          "quest": 0.5974301, ... }
      }
   }
}
```

elastic

Commercial

ELSER

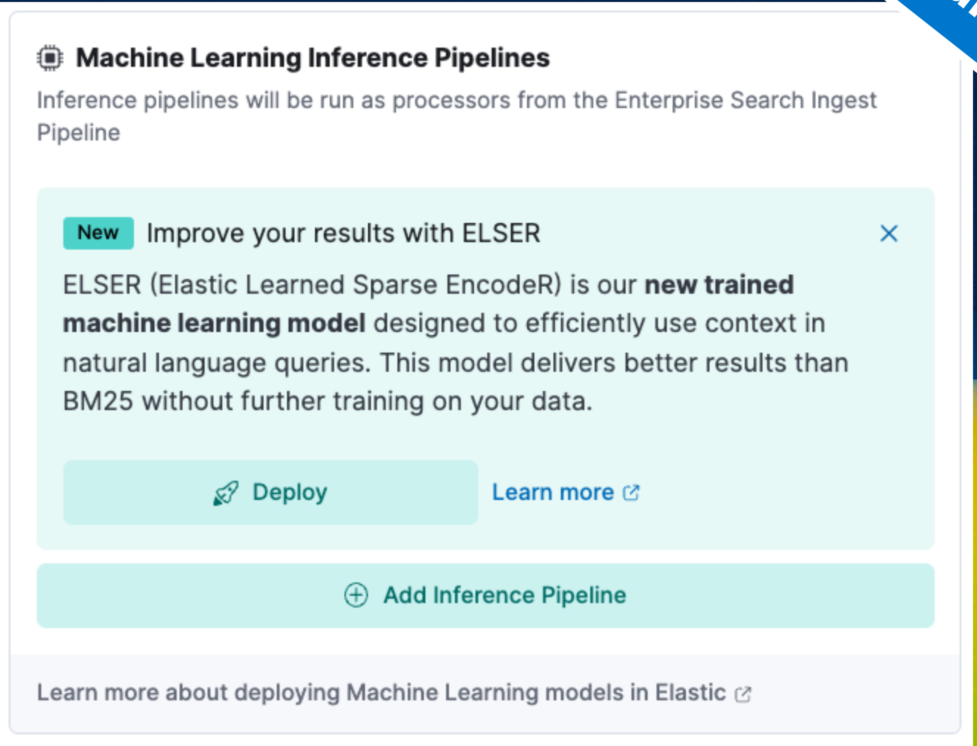
Elastic **L**earned **S**parse Encod**ER**

sparse_vector

Not BM25 or (dense) vector

Sparse vector like BM25

Stored as inverted index



elastic

Hybrid ranking

ranking 1
Term-based score

ranking 2
Dense vector score

ranking 3
Sparse vector score

Reciprocal Rank Fusion (RRF)
blend multiple ranking methods

Combine

Reciprocal Rank Fusion (RRF)

$$RRFscore(d \in D) = \sum_{r \in R} \frac{1}{k + r(d)}$$

D - set of docs
R - set of rankings as permutation on 1..|D|
k - typically set to 60 by default

Dense Vector			
Doc	Score	r(d)	k+r(d)
A	1	1	61
B	0.7	2	62
C	0.5	3	63
D	0.2	4	64
E	0.01	5	65

BM25			
Doc	Score	r(d)	k+r(d)
C	1,341	1	61
A	739	2	62
F	732	3	63
G	192	4	64
H	183	5	65

Doc	RRF Score
A	1/61 + 1/62 = 0,0325
C	1/63 + 1/61 = 0,0323
B	1/62 = 0,0161
F	1/63 = 0,0159
D	1/64 = 0,0156

```
GET index/_search
{
  "retriever": {
    "rrf": {
     "retrievers": [{
          "standard" { "query": {
              "match": {...}
            }
          }
        },{
          "standard" { "query": {
              "sparse_vector": {...}
            }
          }
        },{
          "knn": { ... }
        }
      ]
    }
  }
}
```

Hybrid Ranking

BM25f

+

Sparse Vector

+

Dense Vector

commercial

elastic

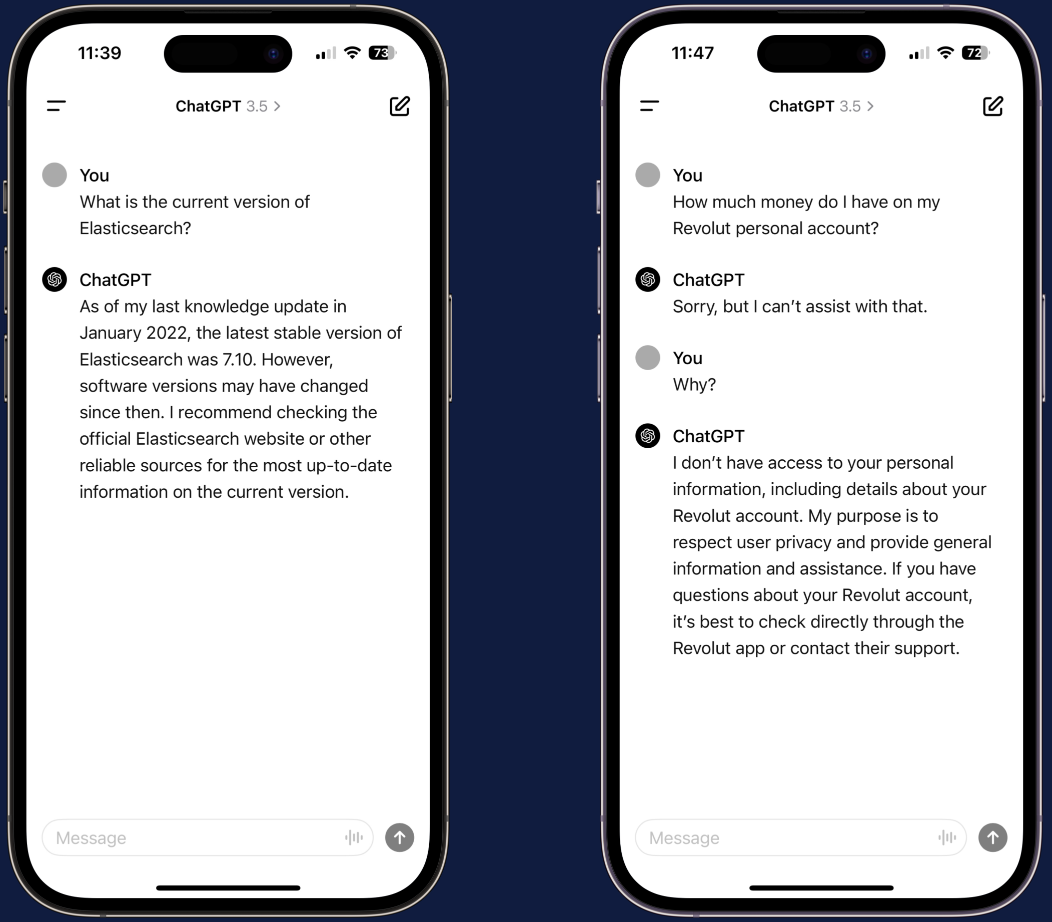
ChatGPT

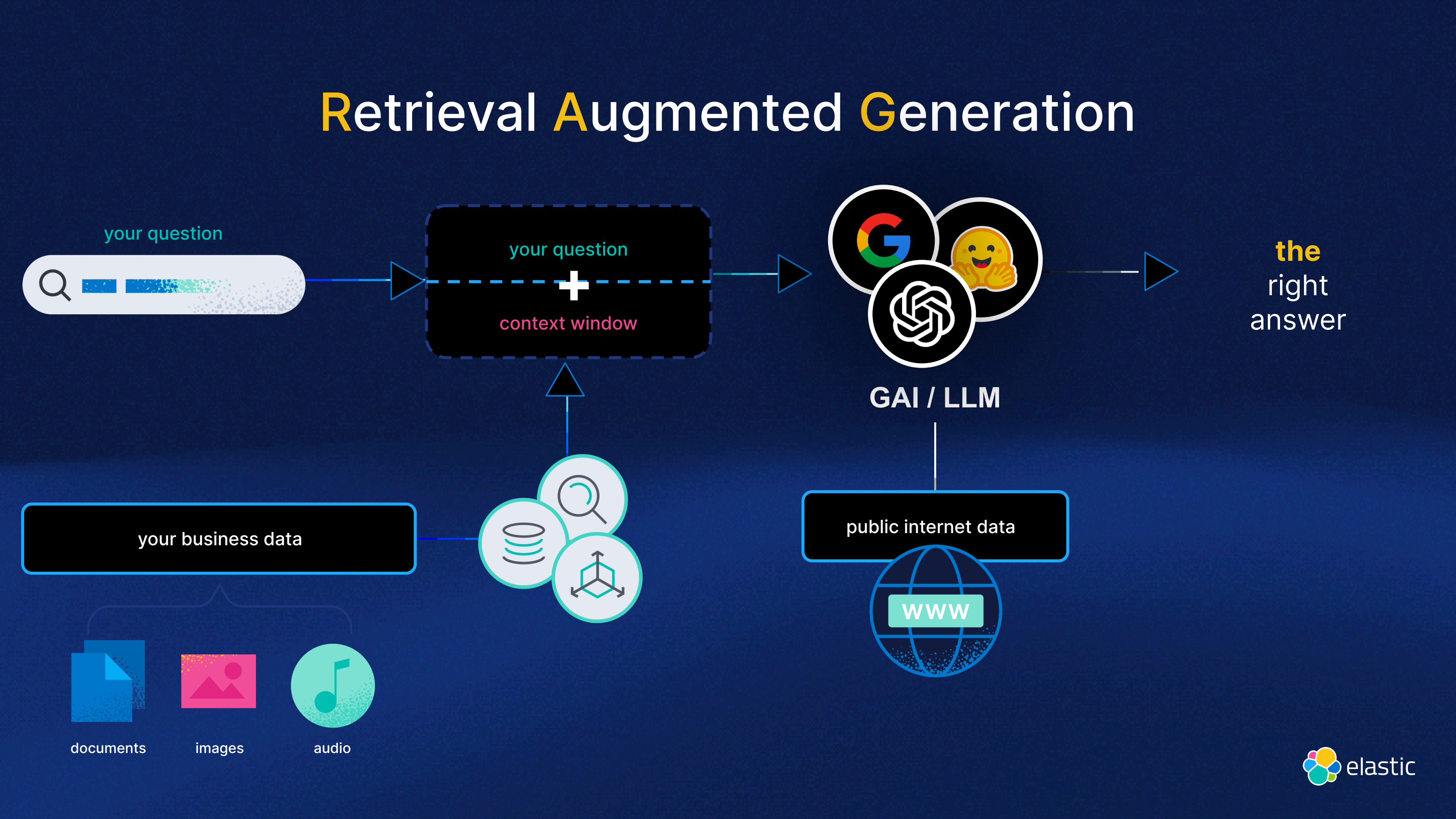
Elastic and LLM


Gen AI
Search engines

LLM: opportunities and limits

elastic


11:39
ChatGPT 3.5 >
You
What is the current version of Elasticsearch?
ChatGPT
As of my last knowledge update in January 2022, the latest stable version of Elasticsearch was 7.10. However, software versions may have changed since then. I recommend checking the official Elasticsearch website or other reliable sources for the most up-to-date information on the current version.
Message
11:47
ChatGPT 3.5 >
You
How much money do I have on my Revolut personal account?
ChatGPT
Sorry, but I can't assist with that.
You
Why?
ChatGPT
I don't have access to your personal information, including details about your Revolut account. My purpose is to respect user privacy and provide general information and assistance. If you have questions about your Revolut account, it's best to check directly through the Revolut app or contact their support.
Message


Retrieval Augmented Generation
your question
your question
+
context window
GAI / LLM
the
right
answer
your business data
public internet data
www
documents
images
audio
elastic

Demo

Elastic Playground

me Home Online banking Enviroment setup

Transaction search Financial summary Customer support

Search your transactions:

This search is not enabled by Elastic and reflects the kind of functionality available to customers today.

Submit

Date	Account	Description	Value	Opening balance	Closing balance
18/06/24	EL03-130981-Transmission	Inbound payment made from EL03-130981-Transmission, St.james's Plac (STJ): 864dce1b-bb95-47d5-87dd-7d02f3b10c3f	7419.0	-825.0	6594.0
18/06/24	EL03-130981-Transmission	Purchase at merchant: Southeastern Grocers, LLC, location: Fayetteville,AR	82.0	6594.0	6512.0
18/06/24	EL03-130981-Transmission	Purchase at merchant: Müller Holding Ltd. & Co. KG, location: Glendale,AZ	188.0	6512.0	6324.0
17/06/24	EL03-130981-Transmission	Payment made from EL03-130981-Transmission to Elwood Erickson, Mitie Grp. (MTO): d37085fc-1382-4593-9cb8-26e5526bd9a0	533.0	20.0	-513.0
17/06/24	EL03-130981-Transmission	Payment made from EL03-130981-Transmission to Classie Johns, Barclays (BARC): 75b603a2-1c1b-45e9-a7ec-4a551bf98a8d	312.0	-513.0	-825.0
16/06/24	EL03-130981-Transmission	Purchase at merchant: E-MART Inc., location: Fayetteville,AR	31.0	51.0	20.0
14/06/24	EL03-130981-Transmission	Purchase at merchant: Dick's Sporting Goods, Inc., location: Montgomery,AL	182.0	329.0	147.0
14/06/24	EL03-130981-Transmission	Purchase at merchant: Valor Holdings Co., Ltd., location: Louisville,KY	96.0	147.0	51.0
13/06/24	EL03-130981-Transmission	Purchase at merchant: The Save Mart Companies, location:	34.0	363.0	329.0

elastic

Elasticsearch

You Know, for Semantic Search

elastic

elastic

Search
a new era

David Pilato
@dadoonet
@pilato.fr