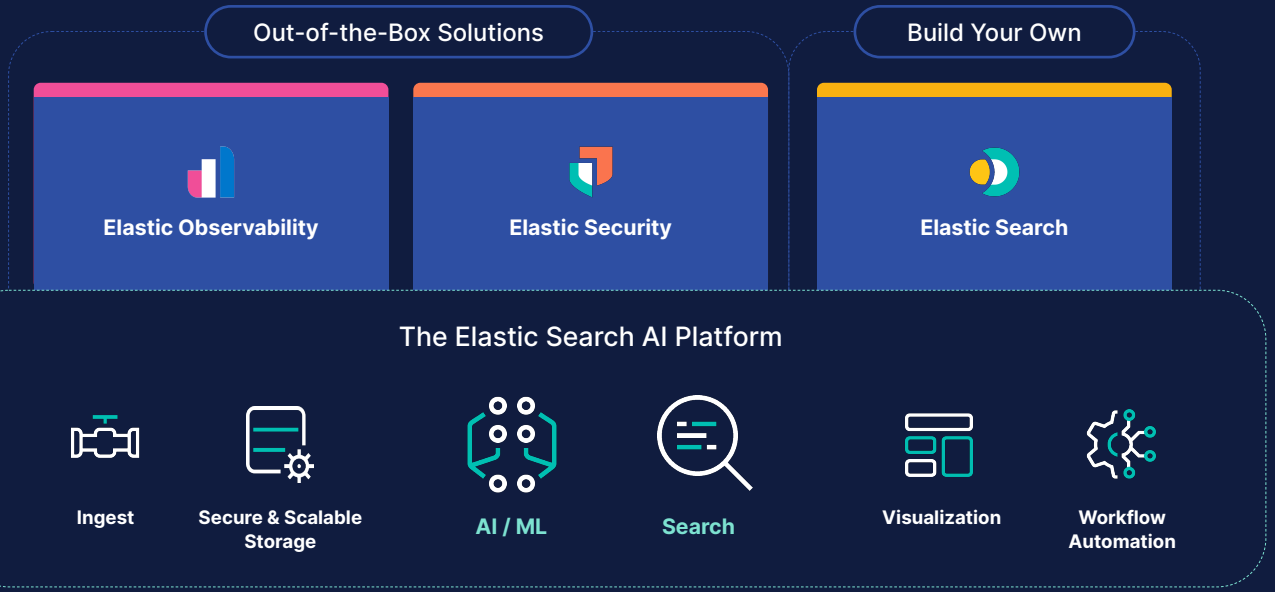
elastic | The Search AI Company

BBL at ekino.

```
$ curl -XPOST https://localhost:9200/speaker/_doc -d '{
  "name" : "David Pilato",
  "jobs" : [
    { "name" : "SRA Europe (SSII)", "date" : "1995" },
    { "name" : "SFR", "date" : "1997"  },
    { "name" : "e-Brands / Vivendi", "date": "2000" },
    { "name" : "DGDDI (douane)", "date" : "2005" },
    { "name" : "elastic", "date" : "2013" }
  ],
  "motivations" : [ "family", "job", "deejay" ],
  "blog" : "https://david.pilato.fr/",
  "twitter" : [ "@dadoonet", "@elasticfr" ],
  "bluesky" : [ "@pilato.fr" ],
  "email" : "david@elastic.co"
}' -H 'Content-Type: application/json'
```

elastic

One Search AI Platform
Two Out-of-the-Box Solutions
The Freedom to Build Anything



elastic

start-local

Elasticsearch — the most widely deployed vector database

Copy to try locally in two minutes

```
curl -fsSL https://elastic.co/start-local | sh
```

Read docs →

OR

Deploy for production

Start free cloud trial

Or, download on-prem

elastic cloud

Elastic pricing

The best way to consume Elastic is Elastic Cloud, a public cloud managed service. Elastic Cloud is available on your preferred cloud provider — AWS, Azure, or Google Cloud. Customers who want to manage the software themselves, whether on public, private, or hybrid cloud, can download the Elastic Stack.

Start free trial Estimate costs →

Standard

As low as
$95 per month[1]

Try free

A great place to start

- Core Elastic Stack features, including security
- Discover, field statistics, Kibana Lens, Elastic Maps, and Canvas
- Alerting and in-stack action

Gold

As low as
$109 per month[1]

Try free

Everything in Standard plus:

- Reporting
- Third-party alerting actions
- Watcher
- Multi-stack monitoring

Platinum

As low as
$125 per month[1]

Try free

Everything in Gold plus:

- Advanced Elastic Stack security features
- Machine learning (ML) – anomaly detection, supervised learning, third-party model management
- Cross-cluster replication

Enterprise

As low as
$175 per month[1]

Try free

Everything in Platinum plus:

- Searchable snapshots
- Support for searchable cold and frozen tiers
- Elastic Maps Server

elastic

serverless

	General purpose	Vector search	Time series
	Best for general search use cases across various data types.	Best for semantic search use cases using vectors with near-real-time retrieval.	Best for retention and analysis of high volume time series data, such as logs of other data streams.
	Try for free	Try for free	**Coming soon**
Ingest Ingest VCUs for indexing data	**$0.14** Per VCU-hour	**$0.14** Per VCU-hour	*Coming soon*
Search Search VCUs for querying data	**$0.09** Per VCU-hour	**$0.09** Per VCU-hour	*Coming soon*
Machine Learning Machine Learning VCUs for trained models and ML jobs	**$0.07** Per VCU-hour	**$0.07** Per VCU-hour	*Coming soon*
Storage Persistent storage in Search AI Lake	**$0.047** Per GB-month	**$0.047** Per GB-month	*Coming soon*
Egress Data transfer out of project 50 GB monthly allowance per organization	**$0.05** Per GB	**$0.05** Per GB	*Coming soon*

elastic

A typical search implementation...

```
CREATE TABLE user
(
   name VARCHAR(100),
   comments VARCHAR(1000)
);
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at french
customs service');
INSERT INTO user VALUES ('David Gageot', 'Engineer at Doctolib');
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

elastic

Search on term

```
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at french
customs service');
INSERT INTO user VALUES ('David Gageot', 'Engineer at Doctolib');
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

```
SELECT * FROM user WHERE name="David";
Empty set (0,00 sec)
```

elastic

Search like

```
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at french
customs service');
INSERT INTO user VALUES ('David Gageot', 'Engineer at Doctolib');
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

```
SELECT * FROM user WHERE name LIKE "%David%";
+--------------+----------------------+
| name         | comments             |
+--------------+----------------------+
| David Pilato | Developer at elastic |
| David Gageot | Engineer at Doctolib |
| David David  | Who is that guy?     |
+--------------+----------------------+
```

elastic

Search for terms

```
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at french
customs service');
INSERT INTO user VALUES ('David Gageot', 'Engineer at Doctolib');
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

```
SELECT * FROM user WHERE name LIKE "%David Pilato%";
+--------------+----------------------+
| name         | comments             |
+--------------+----------------------+
| David Pilato | Developer at elastic |
+--------------+----------------------+
```

elastic

Search with inverted terms

```
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at french
customs service');
INSERT INTO user VALUES ('David Gageot', 'Engineer at Doctolib');
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

```
SELECT * FROM user WHERE name LIKE "%Pilato David%";
Empty set (0,00 sec)
SELECT * FROM user WHERE name LIKE "%Pilato%David%";
Empty set (0,00 sec)
```

elastic

Search for terms

```
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at french
customs service');
INSERT INTO user VALUES ('David Gageot', 'Engineer at Doctolib');
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

```
SELECT * FROM user WHERE name LIKE "%David%" AND
                         name LIKE "%Pilato%";
+--------------+----------------------+
| name         | comments             |
+--------------+----------------------+
| David Pilato | Developer at elastic |
+--------------+----------------------+
```

elastic

Search in two fields

```
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at french
customs service');
INSERT INTO user VALUES ('David Gageot', 'Engineer at Doctolib');
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

```
SELECT * FROM user WHERE name LIKE "%David%" OR
                         comments LIKE "%David%";
+--------------+-----------------------------------------+
| name         | comments                                |
+--------------+-----------------------------------------+
| David Pilato | Developer at elastic                    |
| Malloum Laya | Worked with David at french customs service |
| David Gageot | Engineer at Doctolib                    |
| David David  | Who is that guy?                        |
+--------------+-----------------------------------------+
```

elastic

Search with typos

```
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at french
customs service');
INSERT INTO user VALUES ('David Gageot', 'Engineer at Doctolib');
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

```
SELECT * FROM user WHERE name LIKE "%Dadid%";
Empty set (0,00 sec)
```

elastic

Search with typos

```
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at french
customs service');
INSERT INTO user VALUES ('David Gageot', 'Engineer at Doctolib');
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

```
SELECT * FROM user WHERE name LIKE "%_adid%" OR
                         name LIKE "%D_did%" OR
                         name LIKE "%Da_id%" OR
                         name LIKE "%Dad_d%" OR
                         name LIKE "%Dadi_%";
```

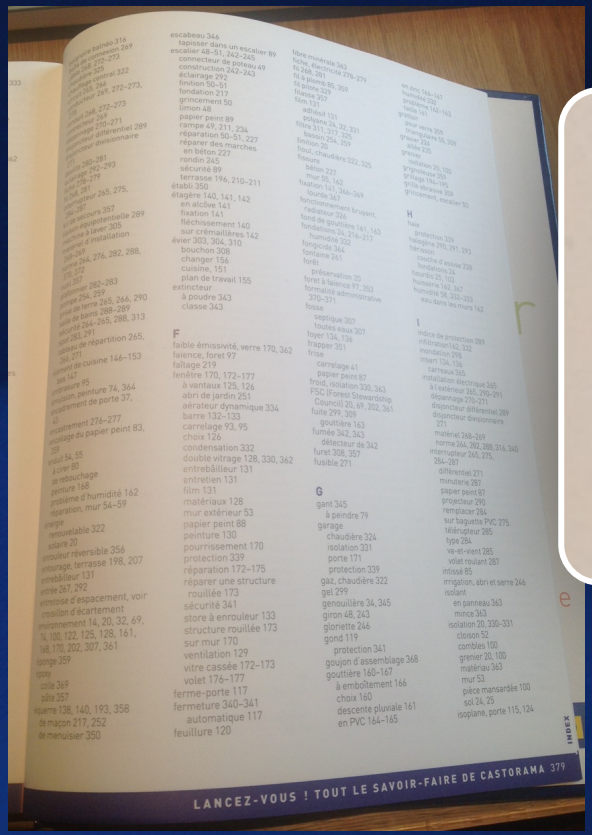
name	comments
David Pilato	Developer at elastic
David Gageot	Engineer at Doctolib
David David	Who is that guy?

elastic

User Interface

elastic

What is a search engine?



- **Index engine** (indexing documents)

F
faible émissivité, verre 170, 362
faïence, foret 97
faîtage 219
fenêtre 170, 172–177
à vantaux 125, 126
abri de jardin 251
aérateur dynamique 334
toutes eaux 307
foyer 134, 136
frapper 351
frise
carrelage 41
papier peint 87
froid, isolation 330, 363
FSC (Forest Stewardship Council) 20, 69, 202, 361
fuite 299, 309

- **Search engine** (within the created indices)

elastic

Demo time!

elastic

Elasticsearch

You Know, for Search

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

elastic

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

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

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

elastic

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

These are not the droids you are looking for.

These
are
not
the
droids
you
are
looking
for

elastic

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

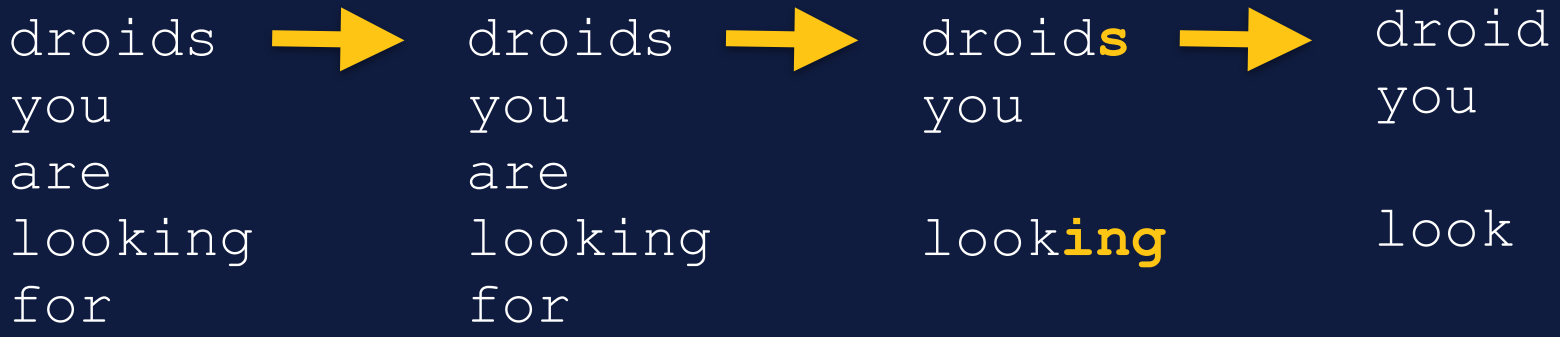
These	these
are	are
not	not
the	the
droids	droids
you	you
are	are
looking	looking
for	for

elastic

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

elastic

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



elastic

```
These are <em>not</em> 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
    }]}
```

Elasticsearch

You Know, for **Vector** Search

elastic

Embeddings represent your data

Example: 1-dimensional vector

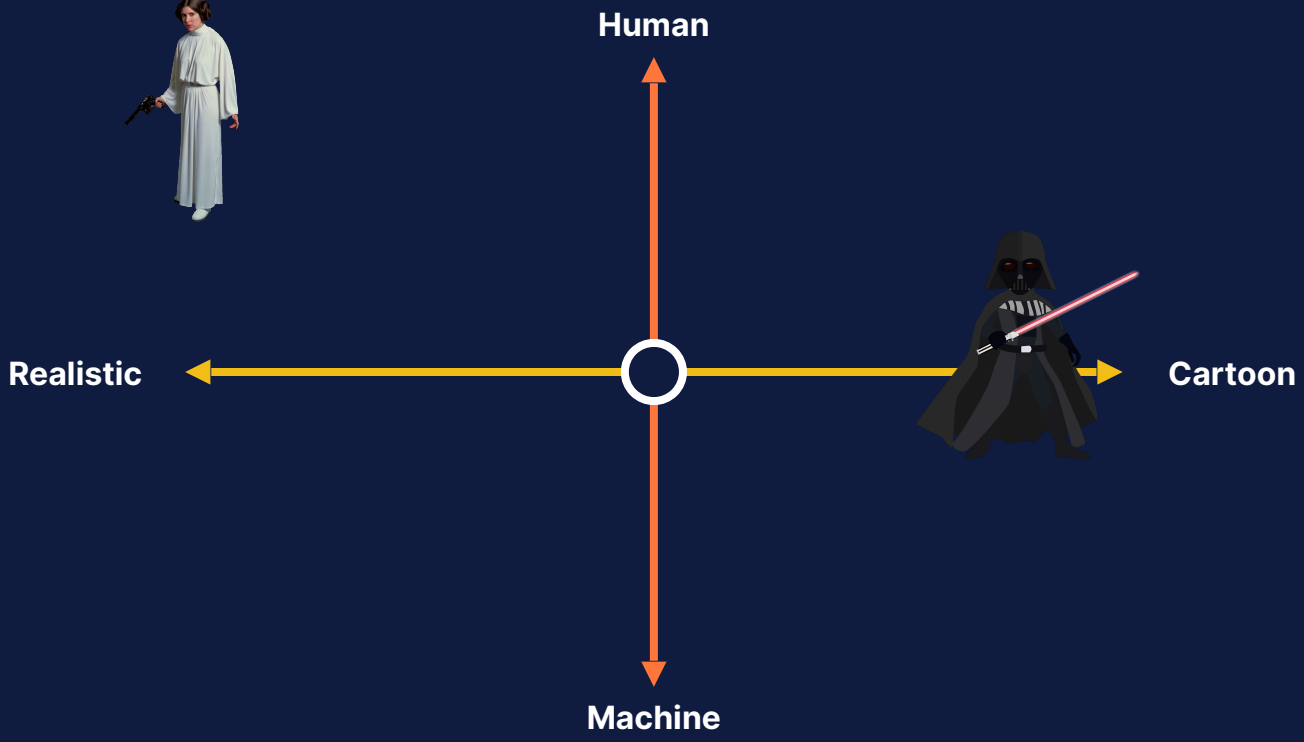
Realistic

Cartoon

Character	Vector
	[-1]
	[1]

elastic

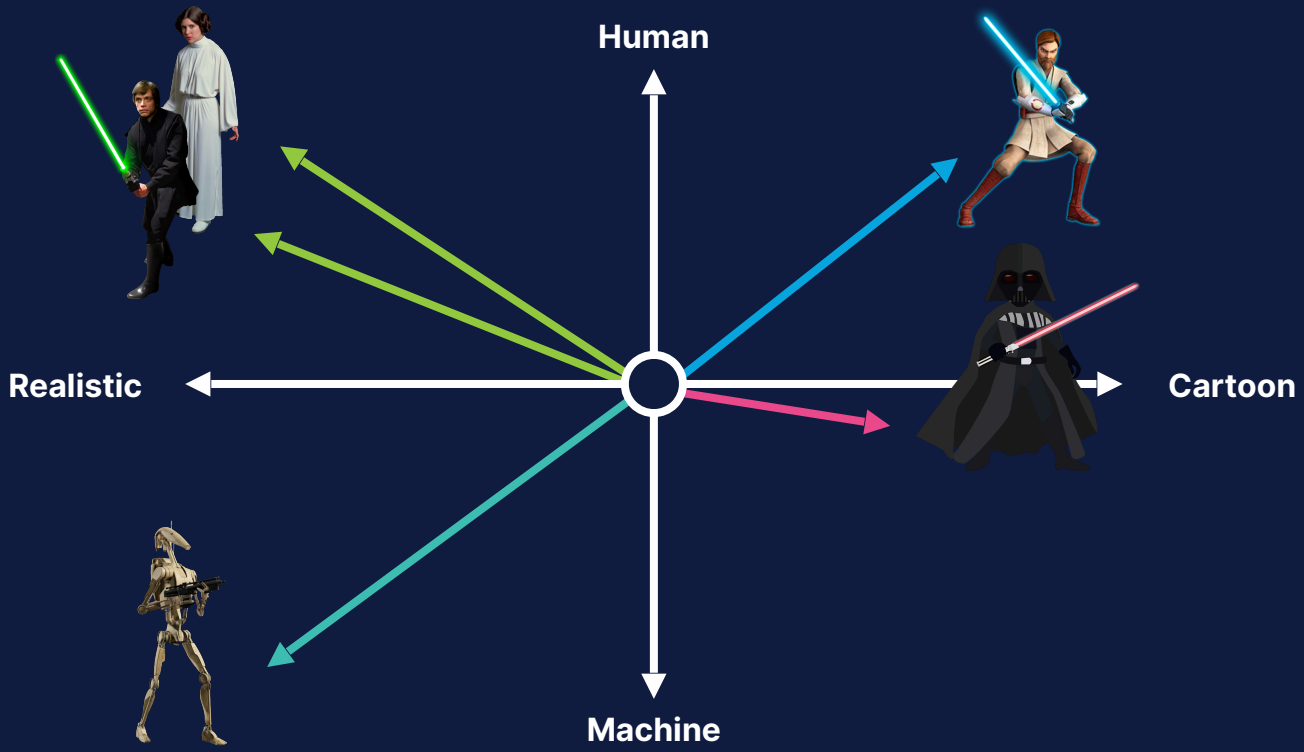
Multiple dimensions
represent different data aspects



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

elastic

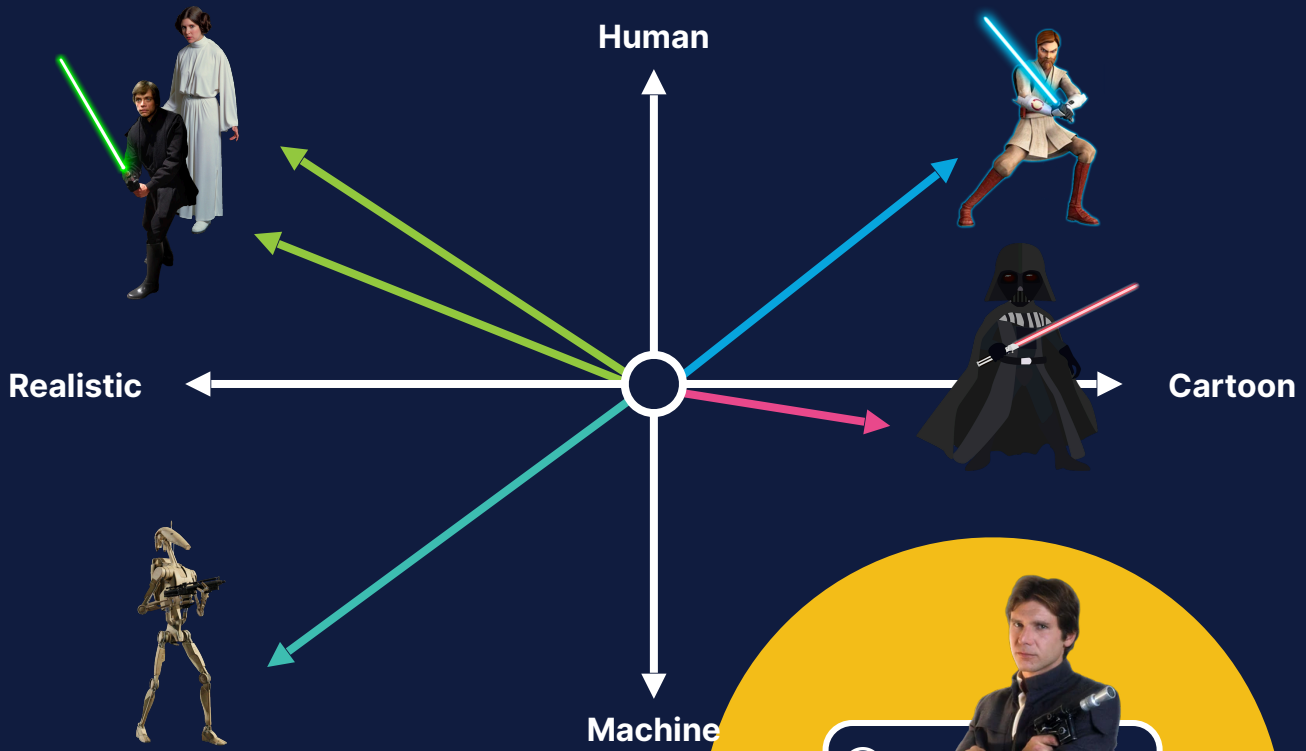
Similar data
is grouped together



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

elastic

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



Rank	Result
Query	
1	
2	
3	
4	
5	

elastic

Data Ingestion and Embedding Generation

Source data

POST /_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,…]
}
```

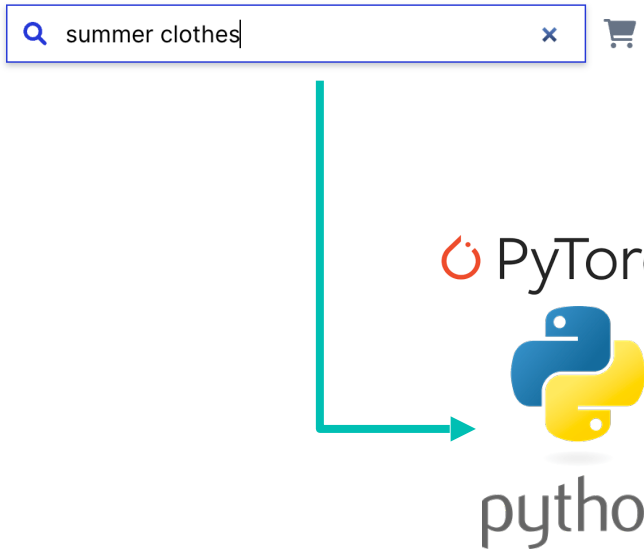
PyTorch

python™

POST /_doc

elastic

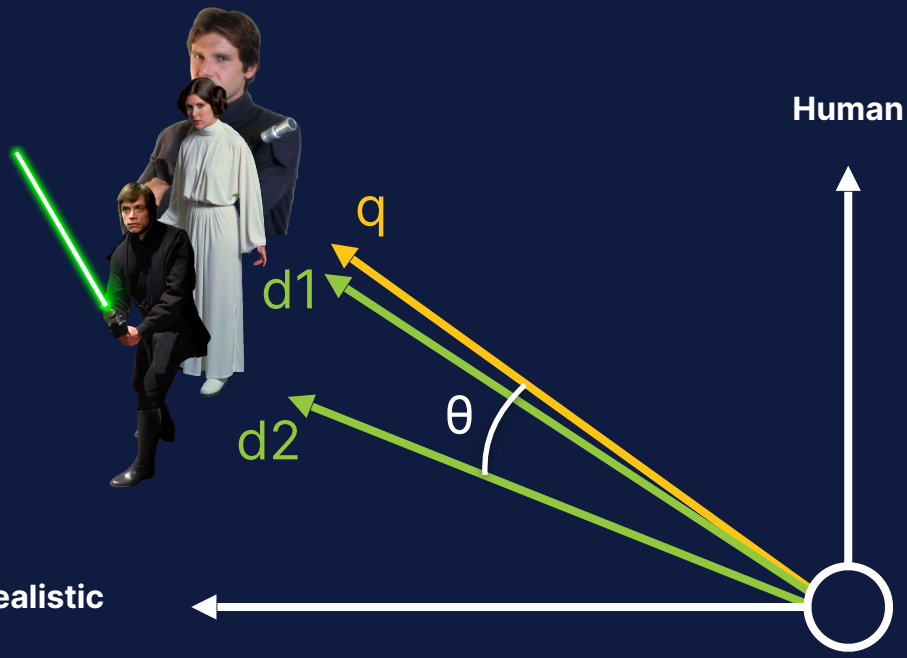
Vector Query




```
GET product-catalog/_search
{
 "query" : {
    "bool": {
      "must": [{
         "knn": {
           "field": "desc_embbeding",
           "num_candidates": 50,
           "query_vector": [0.123, 0.244,...]
         }
      }],
      "filter": {
        "term": {
          "department": "women"
        }
      }
    }
  },
  "size": 10
}
```

elastic

Similarity

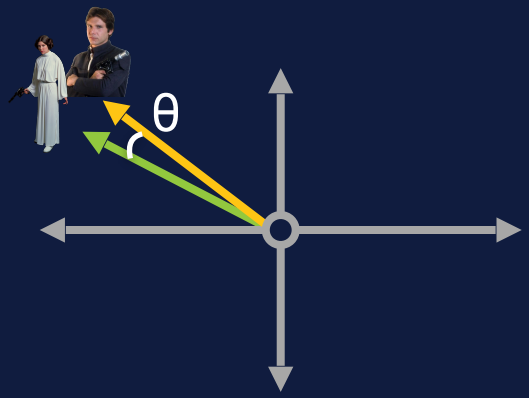


$$cos(\theta) = \frac{\vec{q} \times \vec{d}}{|\vec{q}| \times |\vec{d}|}$$

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

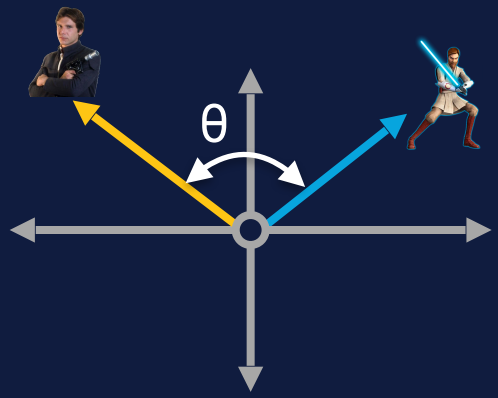
elastic

Similarity: cosine (cosine)



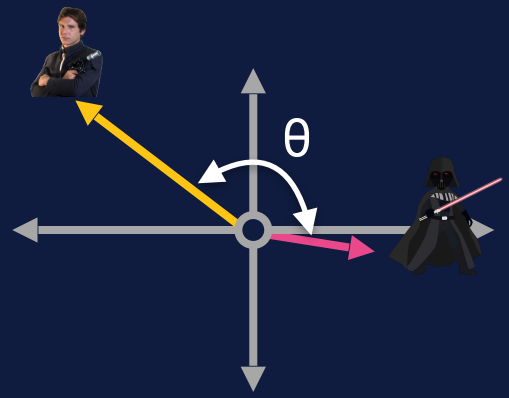
Similar vectors
θ close to 0
cos(θ) close to **1**

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



Orthogonal vectors
θ close to 90°
cos(θ) close to **0**

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

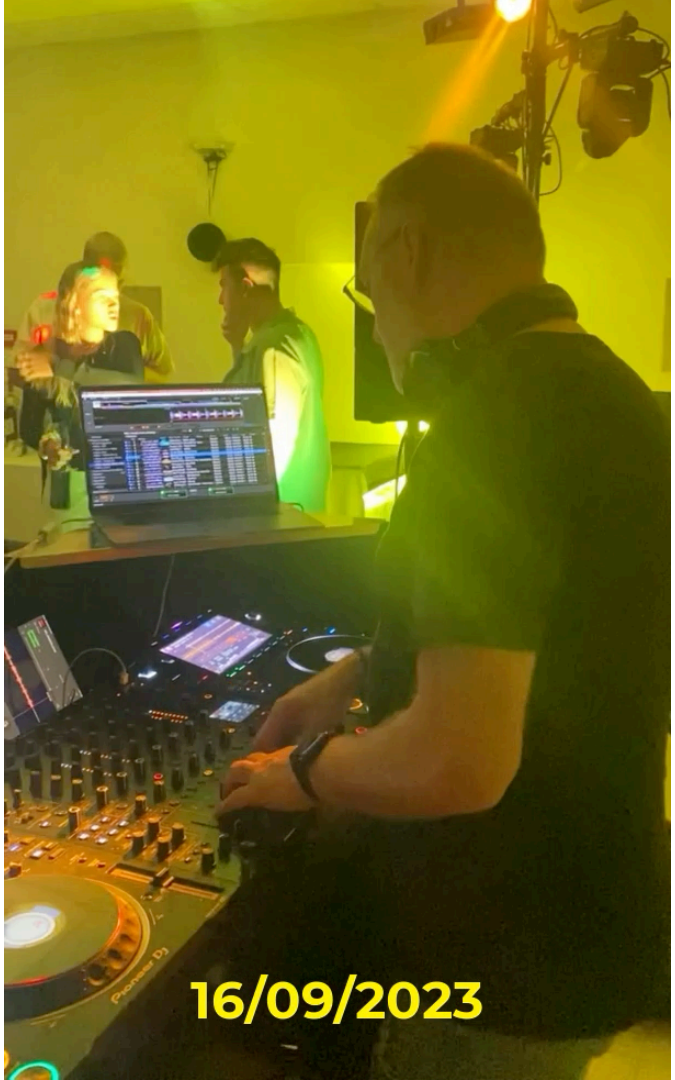


Opposite vectors
θ close to 180°
cos(θ) close to **-1**

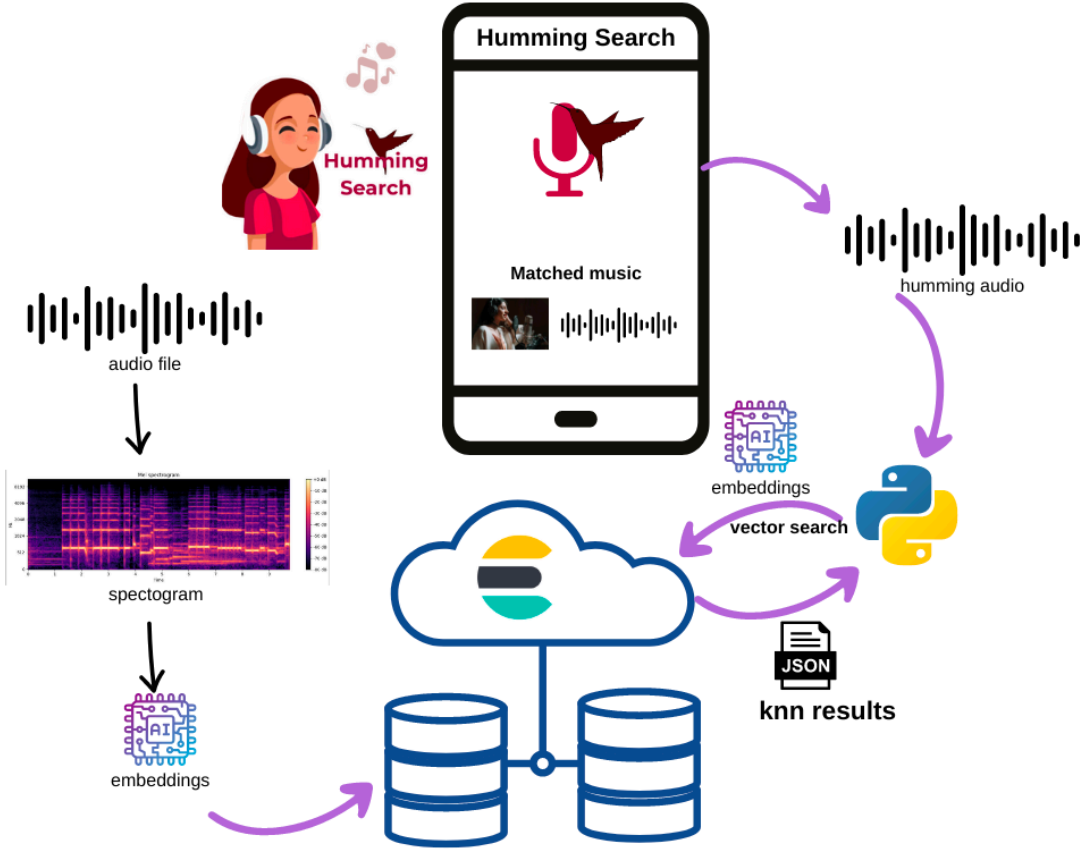
$$\_score = \frac{1-1}{2} = 0$$

elastic

https://djdadoo.pilato.fr/



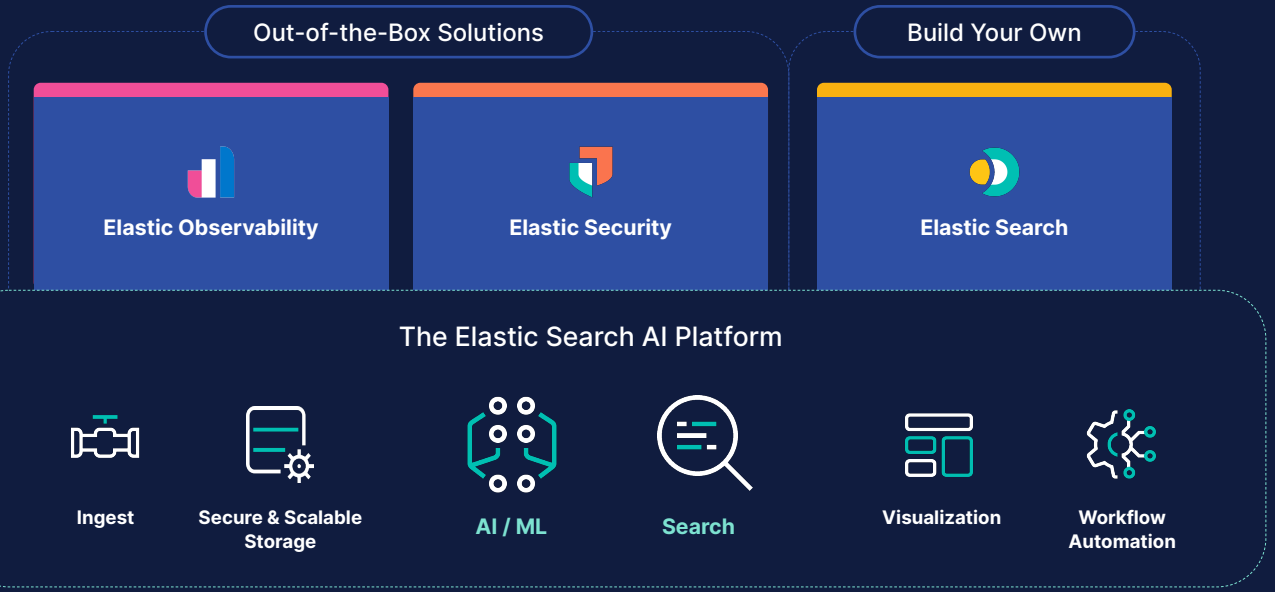
elastic



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

elastic

One Search AI Platform
Two Out-of-the-Box Solutions
The Freedom to Build Anything



elastic

www.meetup.com/ElasticFR

@elasticfr

elastic User Group

discuss.elastic.co

Thank You