🎹 Searching for similar musics 🎻🎸

David Pilato | @dadoonet

Elasticsearch

You Know, for Search

Elasticsearch

elastic

elastic

These are not the droids
you are looking for.

elastic

```
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

```
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 Search

Elasticsearch

You Know, for **Vector** Search

elastic

Embeddings represent your data

Example: 1-dimensional vector

Realistic

Cartoon

Character	Vector
	[-1]
	[1]

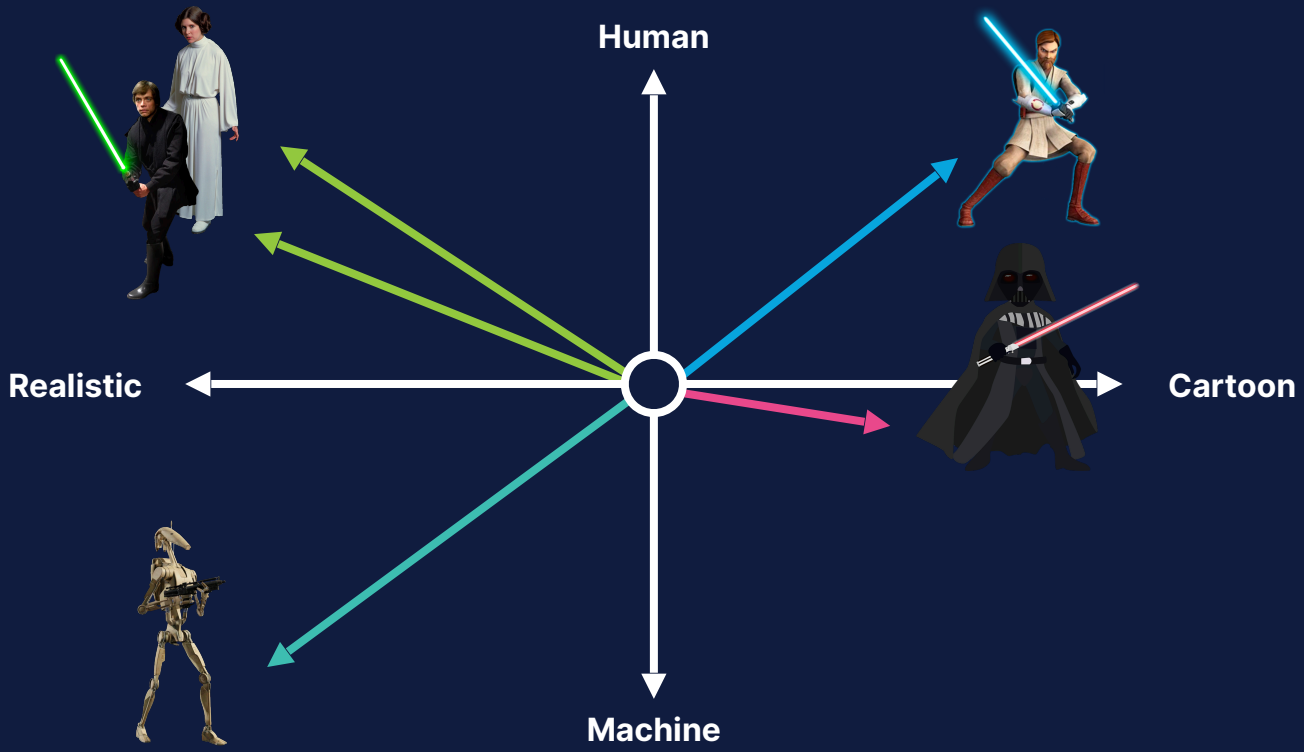
elastic

Multiple dimensions
represent different data aspects

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

elastic

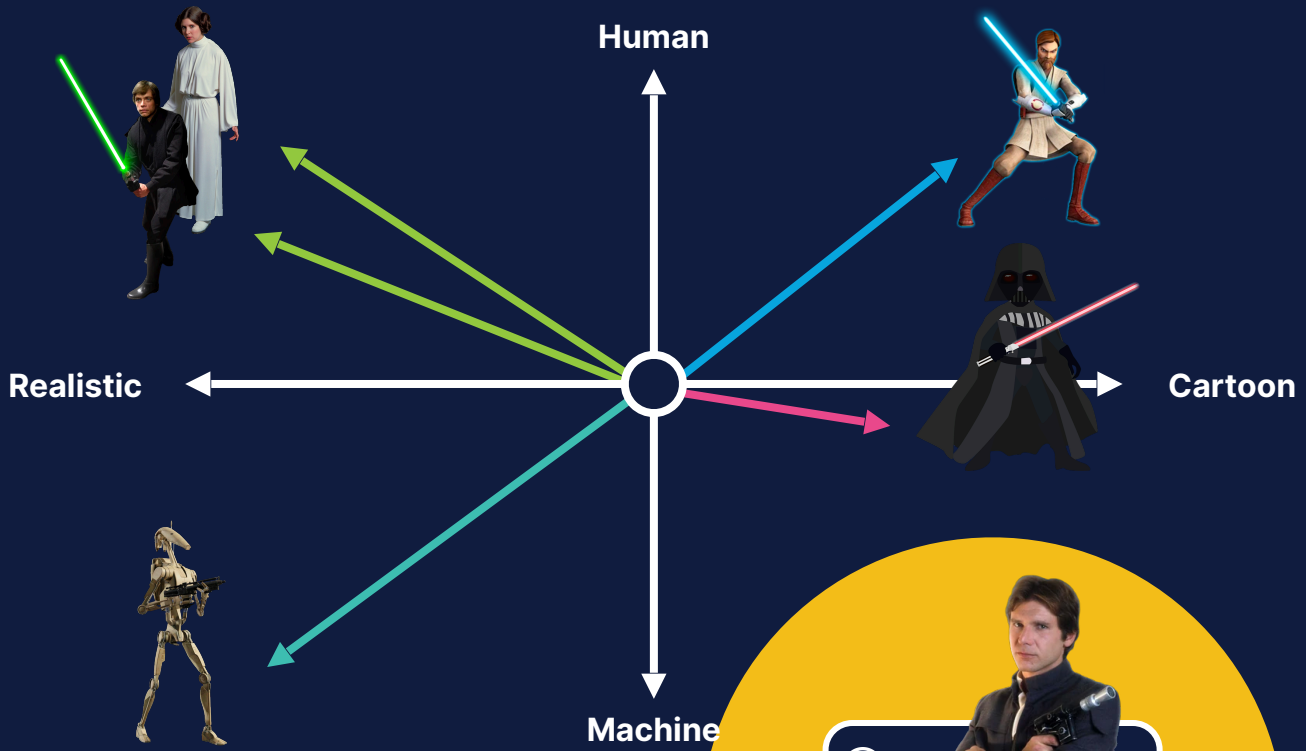
Similar data
is grouped together



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,…]
}
```

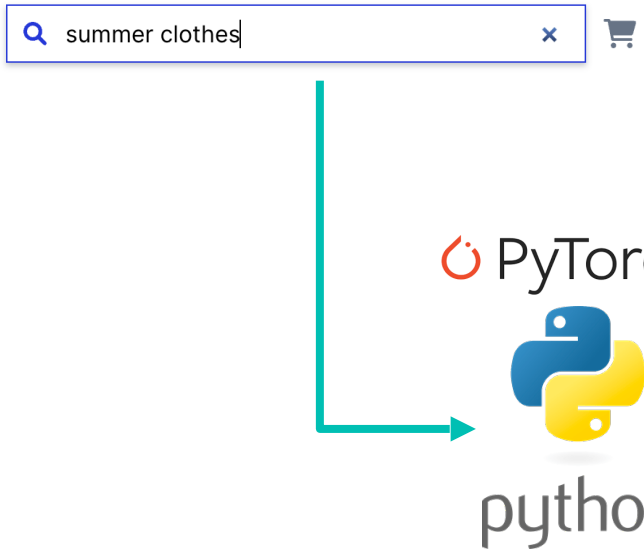
PyTorch

python™

POST /_doc

elastic

Vector Query



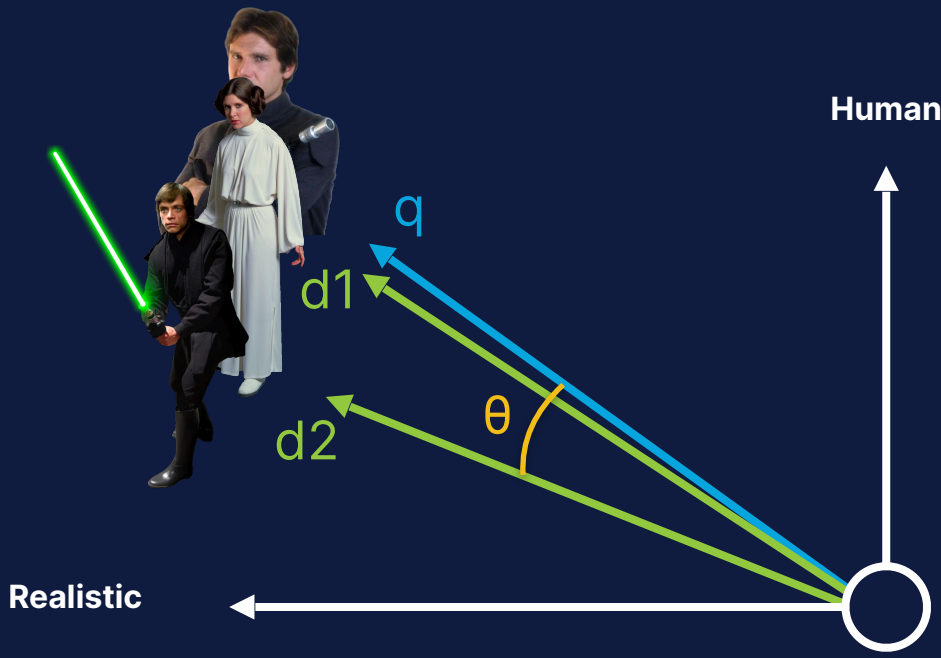

```
GET product-catalog/_search
{
 "query" : {
    "bool": {
      "must": [{
         "knn": {
           "field": "desc_embbeding",
           "num_candidates": 50,
           "query_vector": [0.123, 0.244,...]
         }
      }],
      "filter": {
        "term": {
          "department": "women"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

elastic

But how does it

really work?

Similarity: cosine (cosine)

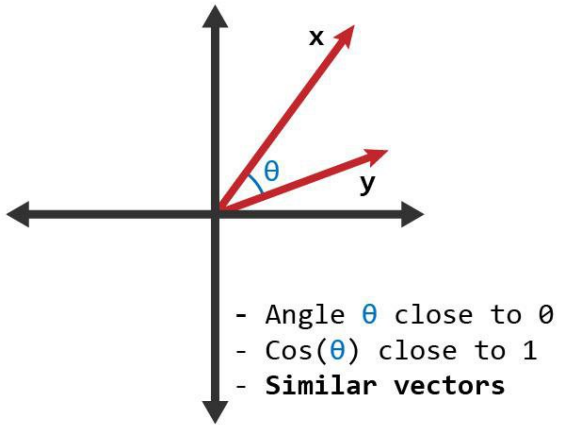


$$cos(\theta) = \frac{\vec{q} \times \vec{d}}{|\vec{q}| \times |\vec{d}|}$$

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

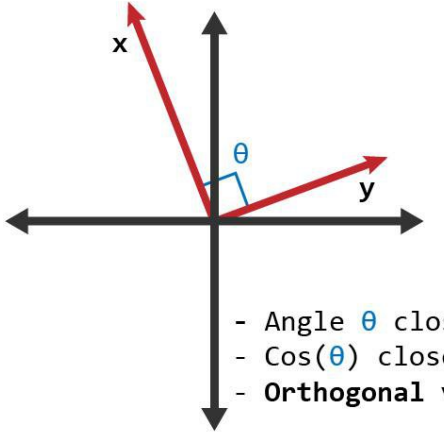
elastic

Similarity: cosine (cosine)



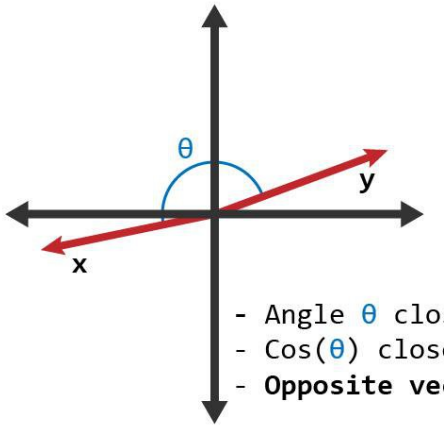
- Angle θ close to 0
- Cos(θ) close to 1
- **Similar vectors**

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



- Angle θ close to 90
- Cos(θ) close to 0
- **Orthogonal vectors**

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

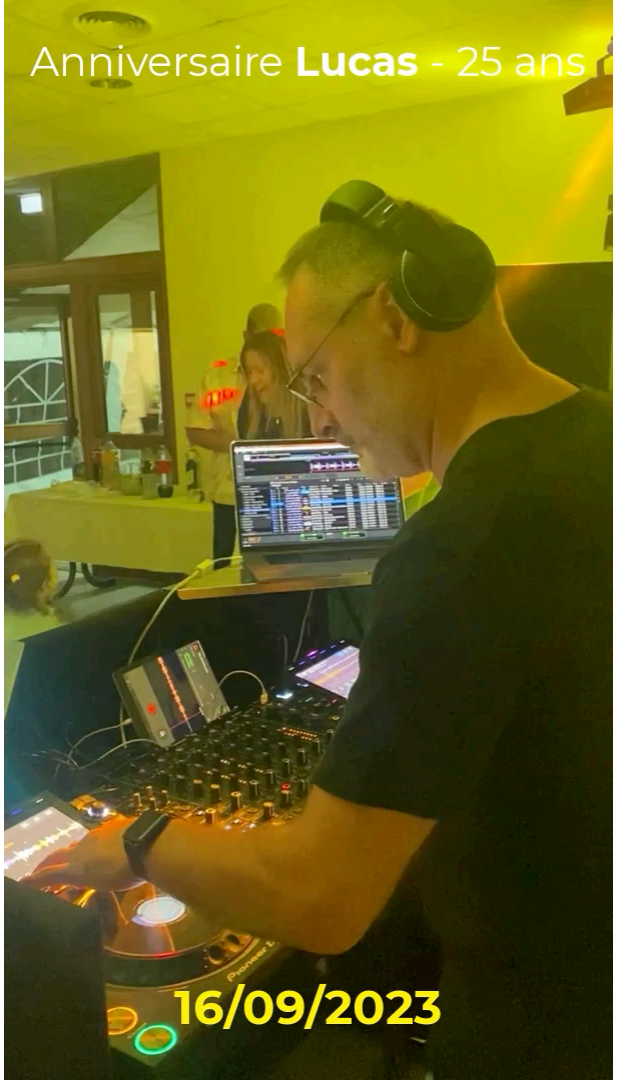


- Angle θ close to 180
- Cos(θ) close to -1
- **Opposite vectors**

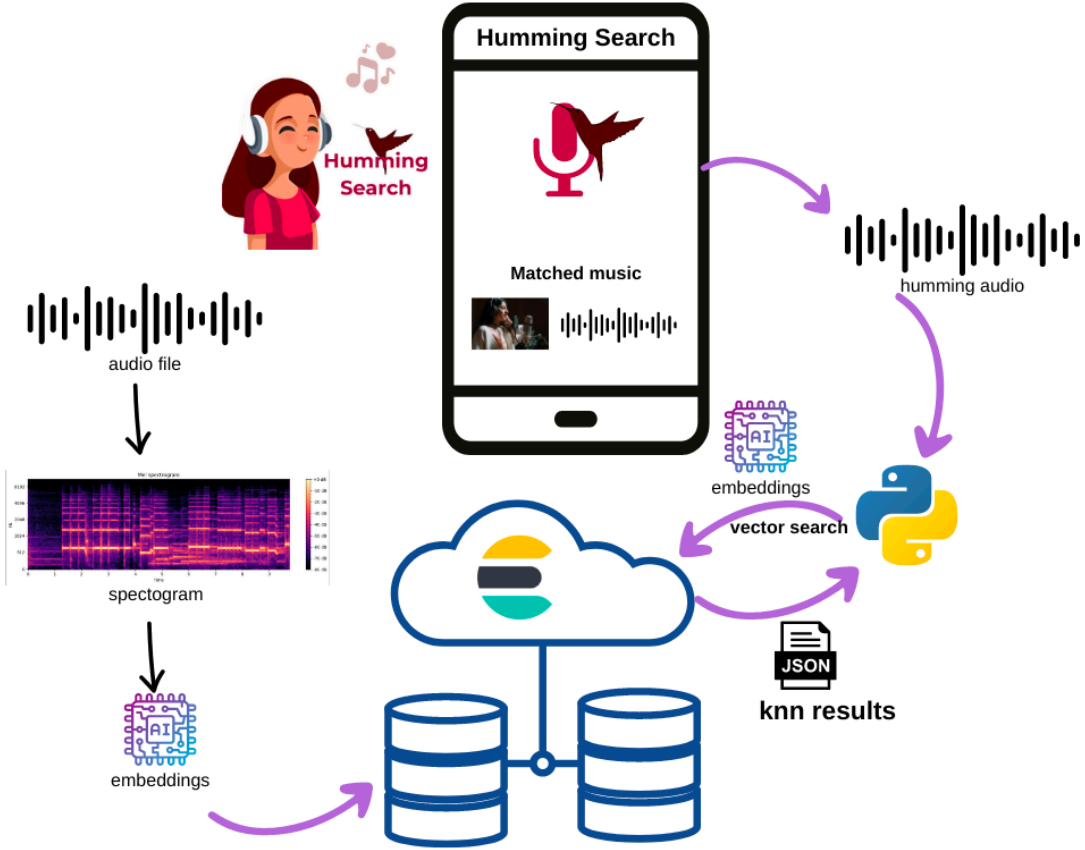
$$\_score = \frac{1-1}{2} = 0$$

elastic

https://djdadoo.pilato.fr/



elastic



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

elastic

🎹 Searching for similar musics 🎻🎸

David Pilato | @dadoonet