



Tutorial: Implementando a Busca de Imagens em banco de dados vetoriais

Alex Salgado
Developer Advocate @ Elastic

- | | |
|--|---|
|  @alexsalgadoprof |  @alexsalgadoprof |
|  salgado |  /in/alex-salgado/ |





Alex Salgado
Senior Developer
Advocate LATAM

 @alexsalgadoprof

 salgado

 @alexsalgadoprof

 /in/alex-salgado/

- **Mestre** em Ciência da Computação pela UFF (Games)
- **MBA** UFF
- **PhD Candidate UFF: Robótica/Visão Computacional**

- + 25 anos de experiência na área de desenvolvimento de software
- Ocupei diversos cargos, trabalhando em **startups**, pequenas e grandes empresas como Oracle, CSN, BRQ/IBM, **Chemtech/Siemens (9 anos)**.
- 8 anos como professor universitário



Preocupações em torno da IA Generativa.

80%

Dados mundiais são não-estruturados

[KPMG Generative AI Survey](#)

[The Prompt: Generative AI survey | Google Cloud Blog](#)



Three solutions powered by one stack

3 solutions



Enterprise Search



Observability



Security

Powered by
the Elastic Stack

Kibana

Elasticsearch

Agent

Beats

Logstash

Deployed
anywhere



Elastic Cloud



Elastic Cloud
Enterprise



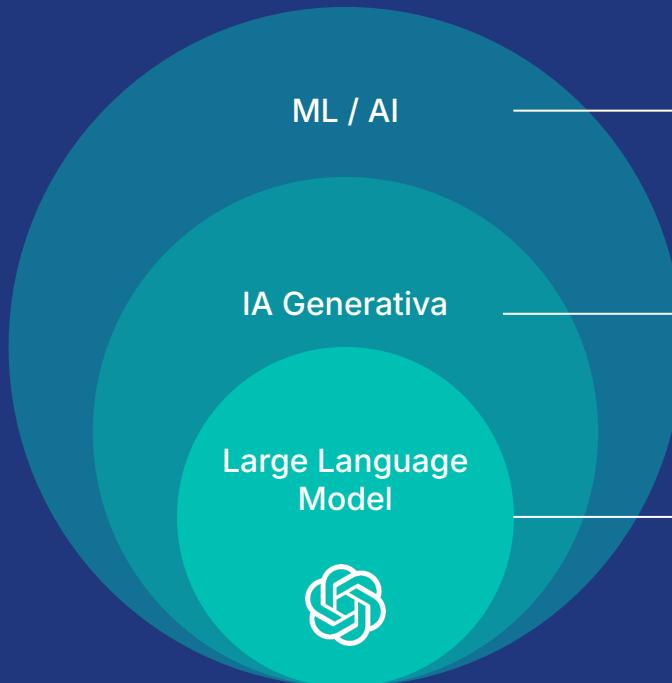
Elastic Cloud
on Kubernetes

SaaS

Orchestration



Conceitos básicos de ML, IA Generativa e LLMs



O que?

Algoritmos programados para aprender o comportamento dos dados e fazer previsões

Casos de uso

Detecção de anomalias, forecasting, reconhecimento de imagem, PLN

Algoritmos programados para criar novos dados

Chatbots, geradores de texto, imagem e música

Algoritmos (Deep Learning) treinados com grandes volumes de dados e programados para criar novos dados

Chatbots, geradores de texto, tradutores, geradores de código, aplicativos de pergunta e resposta

Blog referência

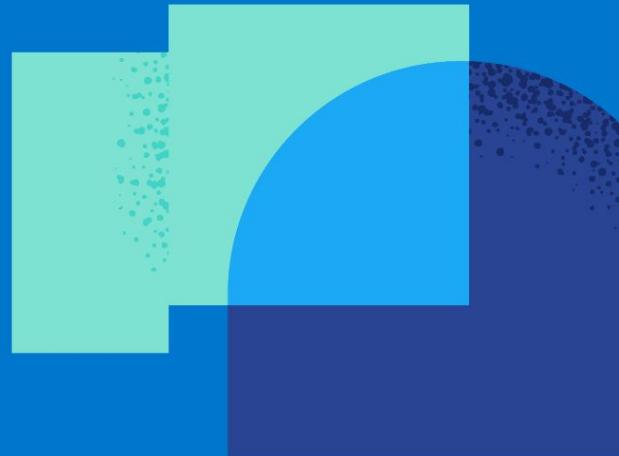
<https://www.elastic.co/search-labs/finding-your-puppy-with-image-search>



Finding your puppy with Image Search

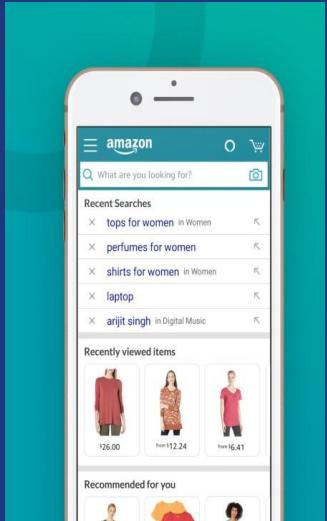
Have you ever been in a situation where you found a lost puppy on the street and didn't know if it had an owner? Learn ho...

November 7, 2023 • Alex Salgado

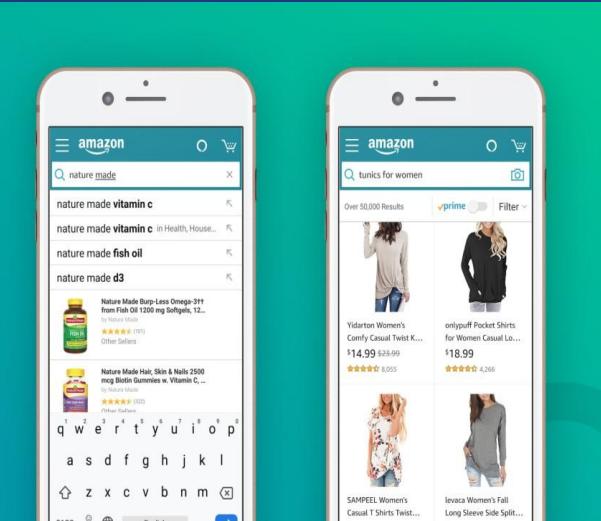


Elasticsearch: You Know, for Search

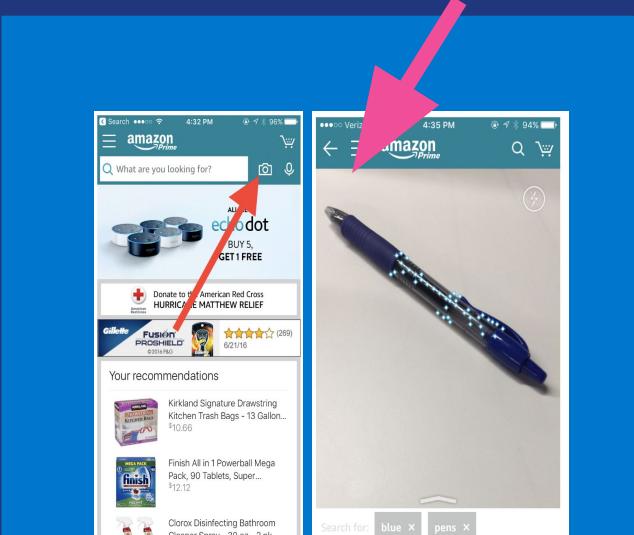
As pessoas estão mudando o jeito de pesquisar



1-Busca textual



2-Busca textual + Semantica



3-Busca por Imagem e Som



How fast should my internet be?

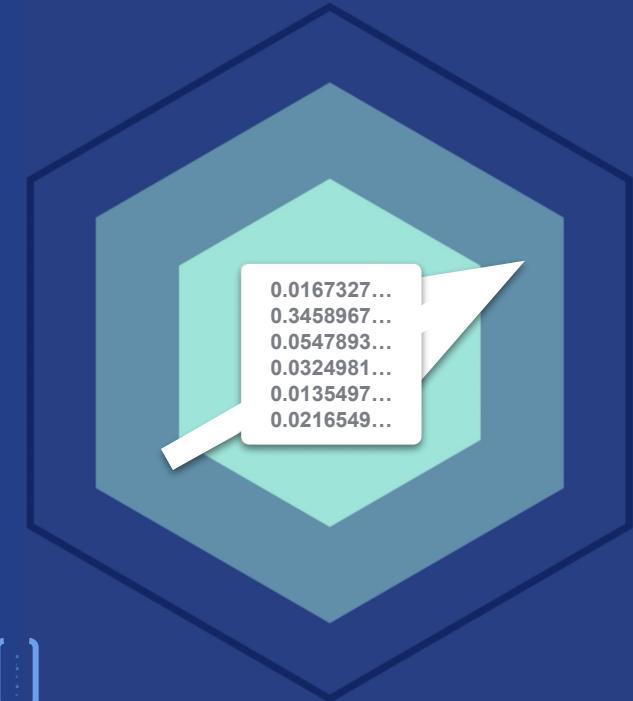
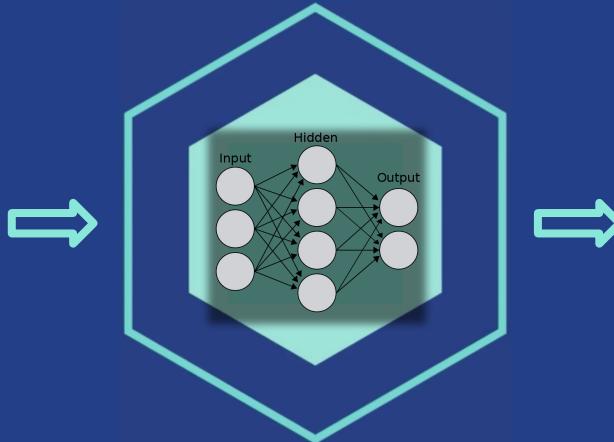
In order to stream from our service you will need a high quality connection. The required connection speed for using the service will vary depending on the quality of

you wish
vice. For

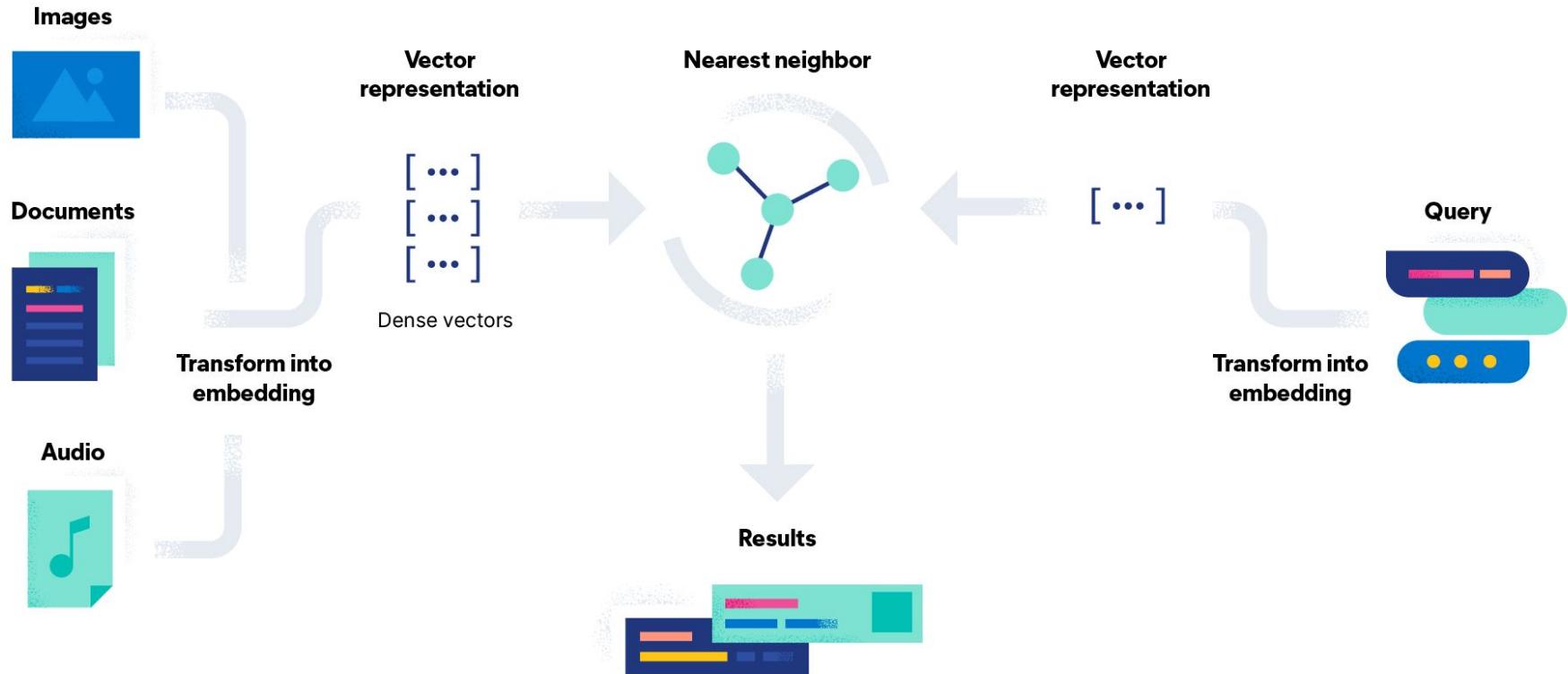
recommend at least...

O que é similaridade de vetores?

Converte dados em representações vetoriais onde as distâncias representam similaridade.



Queries are also vectorized



What is a Vector?

Embeddings represent your data

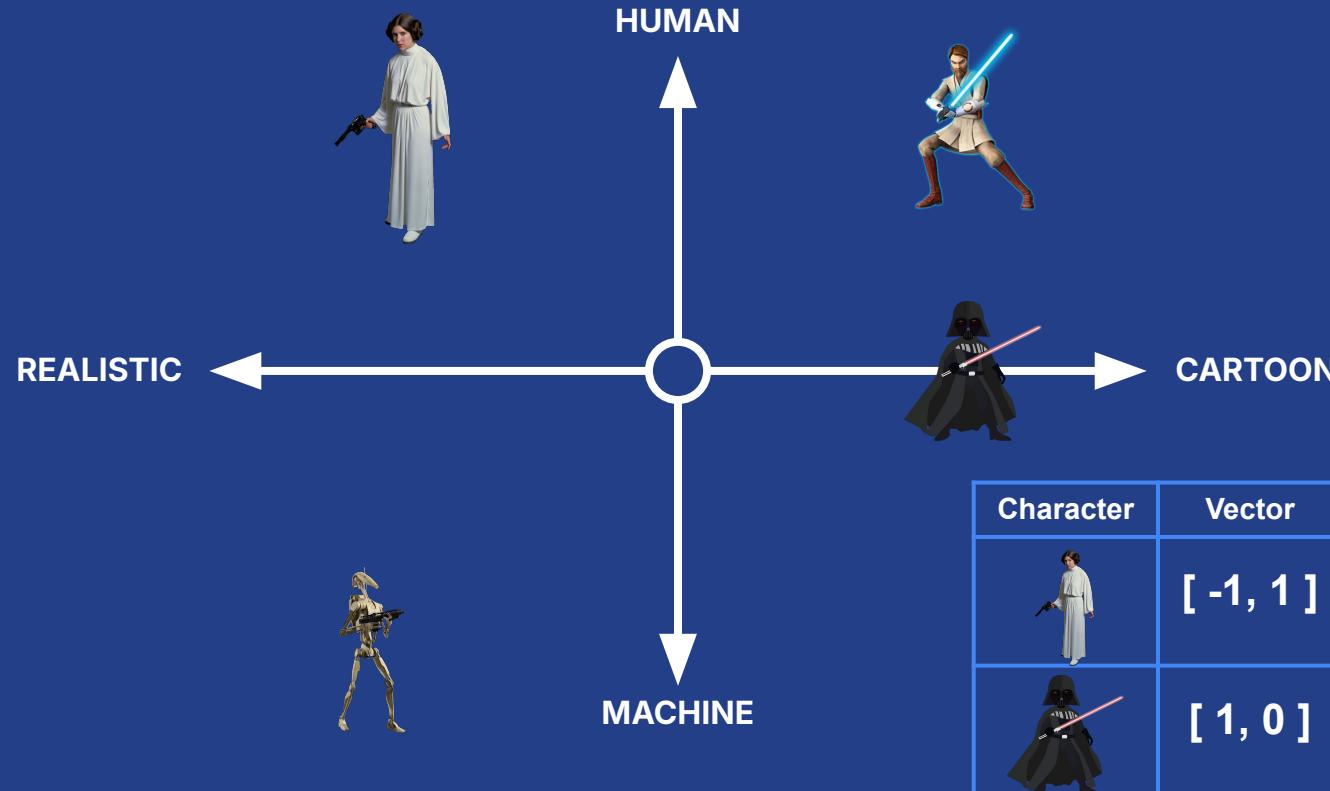
Example: 1-dimensional vector



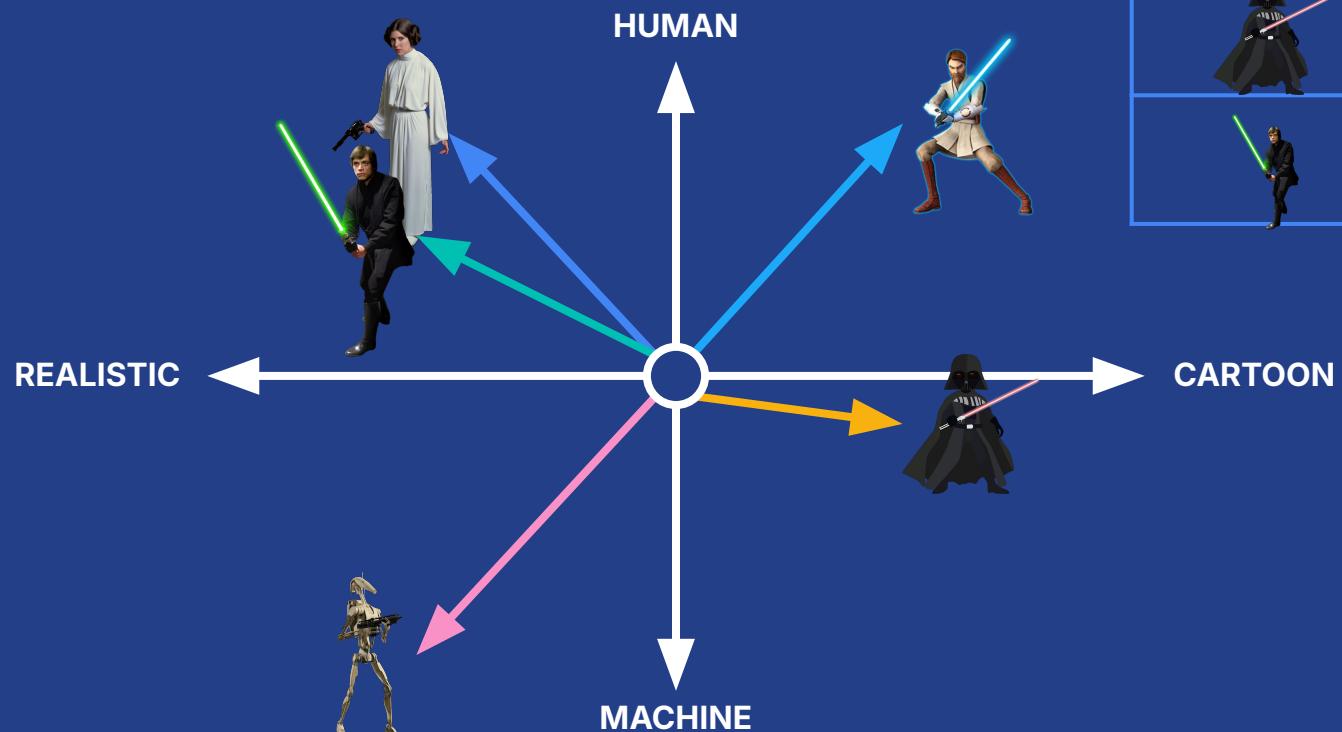
REALISTIC ← → CARTOON

Character	Vector
	[-1]
	[1]

Multiple dimensions represent different data aspects

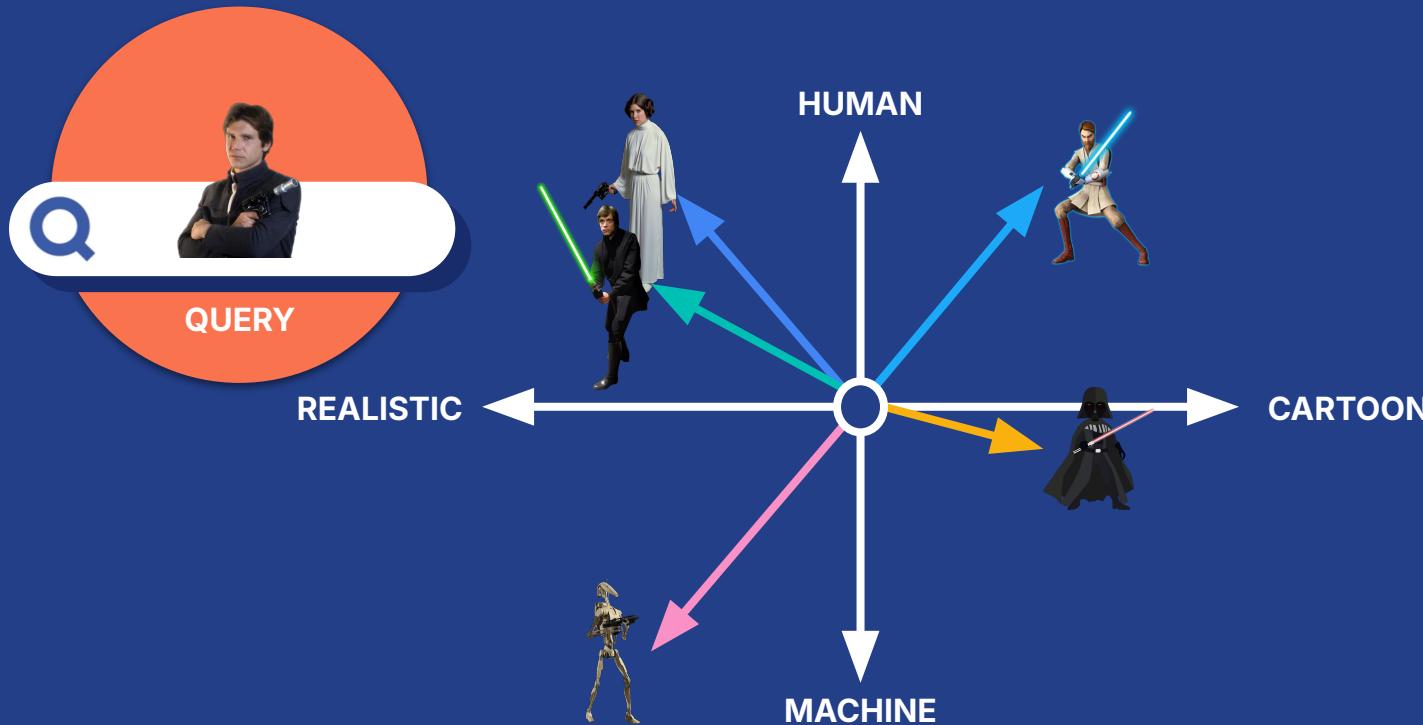


Similar data is grouped together



Character	Vector
	[-1.0, 1.0]
	[1.0, -0.1]
	[-1.0, 0.8]

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



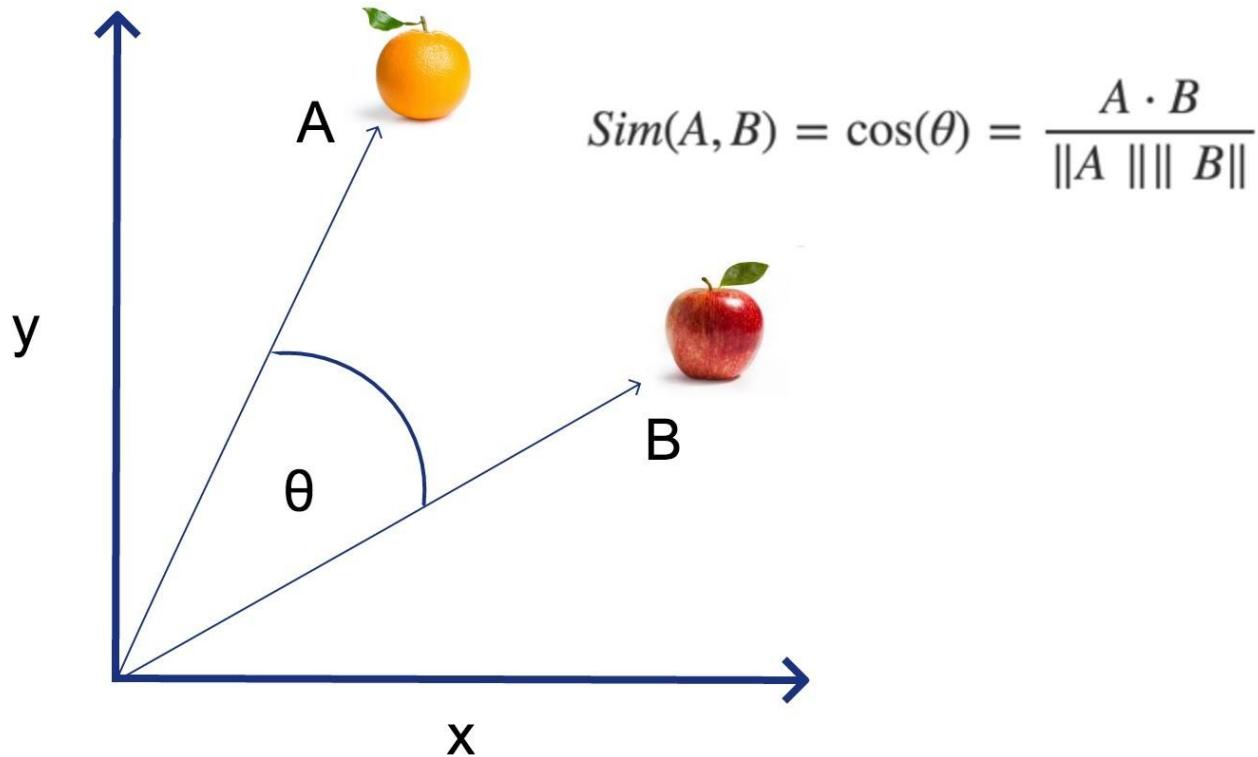
Relevance	Result
Query	
1	
2	
3	
4	
5	



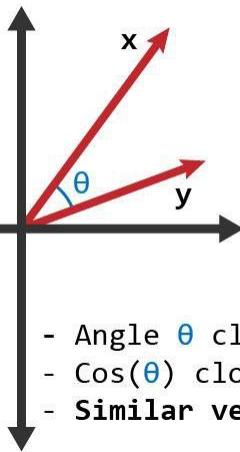
Como essa busca por similaridade realmente funciona?



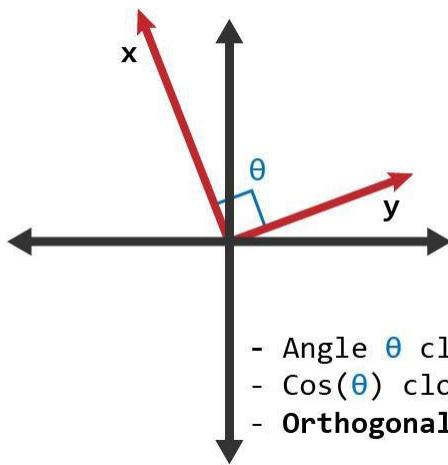
Similarity: Cosine



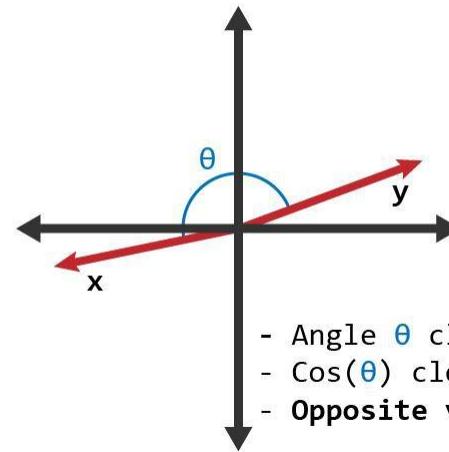
Similarity: Cosine



- Angle θ close to 0
- $\cos(\theta)$ close to 1
- **Similar vectors**

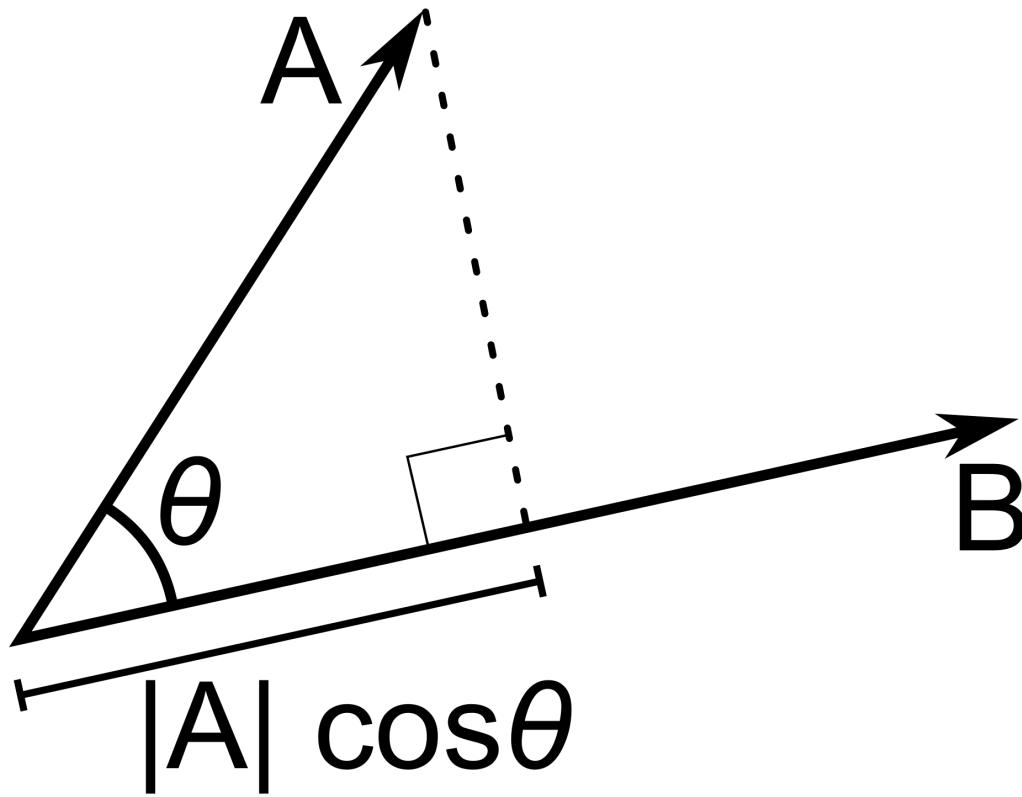


- Angle θ close to 90
- $\cos(\theta)$ close to 0
- **Orthogonal vectors**

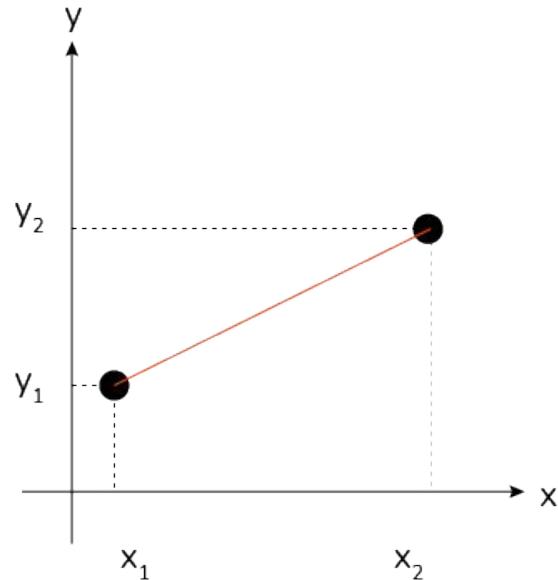


- Angle θ close to 180
- $\cos(\theta)$ close to -1
- **Opposite vectors**

Similarity: Dot Product



Similarity: Euclidean / L2



Demo

<https://www.elastic.co/search-labs/blog/implementing-image-search-with-elasticsearch>



Finding your puppy with Image Search

Have you ever been in a situation where you found a lost puppy on the street and didn't know if it had an owner? Learn ho...

November 7, 2023 • Alex Salgado

Código no github

<https://github.com/salgado/image-search-01/blob/main/finding-your-puppy-with-image-search.ipynb>



Facilidades para o DEV

elastic

Find apps, content, and more.

Setup guides

Endpoints & API keys

Search Content Elasticsearch indices dog-image-index

dog-image-index

View in Playground

Overview Documents Index mappings Pipelines

Ingestion type API

Document count 11

Getting started with Elastic API

Select your client

Elastic builds and maintains clients in several popular languages and our community has contributed many more. Select your favorite language client or dive into the [Console](#) to get started.

Choose one

cURL Python JS PHP Go Ruby

Learn more

Elasticsearch clients Run API requests in Console

Try it now in Console

With Console, you can get started right away with our REST APIs. No installation required.

Try Console now

Install a client

Python

Copy

Notebooks

Console

Enterprise Search App Search

elasticsearch

Introdução aos Bancos de Dados Vetoriais

O que é Busca Vetorial?

- A busca vetorial permite realizar pesquisas com base em similaridade entre vetores em vez de palavras-chave, ideal para imagens e textos.

Casos de Uso Comuns:

- Busca por produtos em e-commerce, reconhecimento facial, e, como veremos, encontrar cães perdidos por meio de imagens.

Case Prático:

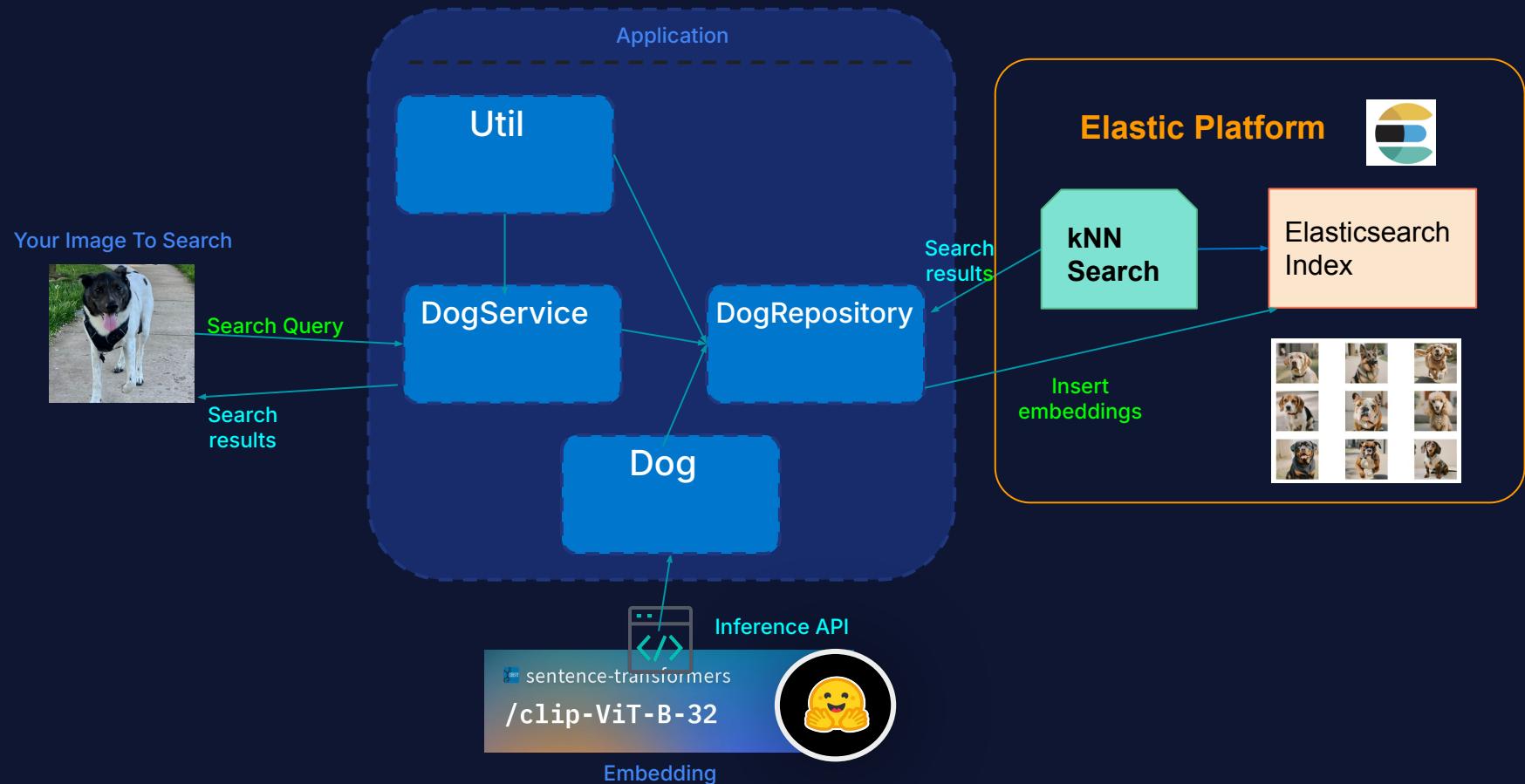
- Encontrando o cãozinho Luigi, perdido durante um passeio, usando uma foto para realizar uma busca no banco de dados vetorial.

Arquitetura da Solução

- **Componentes Principais:**
 - **Elasticsearch:** Usado para armazenar e buscar embeddings (vetores).
 - **Jupyter Notebook (Google Colab):** Ambiente para implementação e testes.
 - **Modelo de Embeddings:** Usamos o **CLIP-ViT-B-32** para converter imagens em vetores.
- **Diagrama da Arquitetura:**
 - **Etapas:**
 1. Extração de embeddings das imagens.
 2. Armazenamento dos vetores no Elasticsearch.
 3. Consulta no banco vetorial via busca de similaridade.

Image Search Architecture

Generate embeddings outside Elasticsearch



Preparando o Ambiente

- **Ferramentas e Pacotes Necessários:**

Clonando o repositório e instalando dependências no Jupyter Notebook:

```
!git clone https://github.com/salgado/image-search-01.git
```

```
!pip -q install Pillow sentence_transformers elasticsearch
```

Conectando-se ao Elasticsearch e configurando o índice:

```
from elasticsearch import Elasticsearch  
es = Elasticsearch(cloud_id="seu-cloud-id", api_key="sua-api-key")
```

Classe Util e Criação do Índice

- **Função da Classe Util:**
 - **Gerenciamento do índice no Elasticsearch:** Criação e exclusão do índice.
- **Diagrama de Função:**
 - A função `create_index()` cria o índice no Elasticsearch e define as configurações necessárias para o armazenamento dos vetores.
- *(diagrama de indexação de imagens, cada imagem tem um ID, embedding e metadados associados, como o nome do cão e dono)*

Exemplo de Código:

```
Util.create_index(es, "dog-image-index")
```

Classe Dog – Representando o Cão

Atributos e Funções:

- A classe `Dog` define o cachorro, contendo atributos como ID, caminho da imagem, raça, nome do dono, e embedding gerado a partir da imagem.

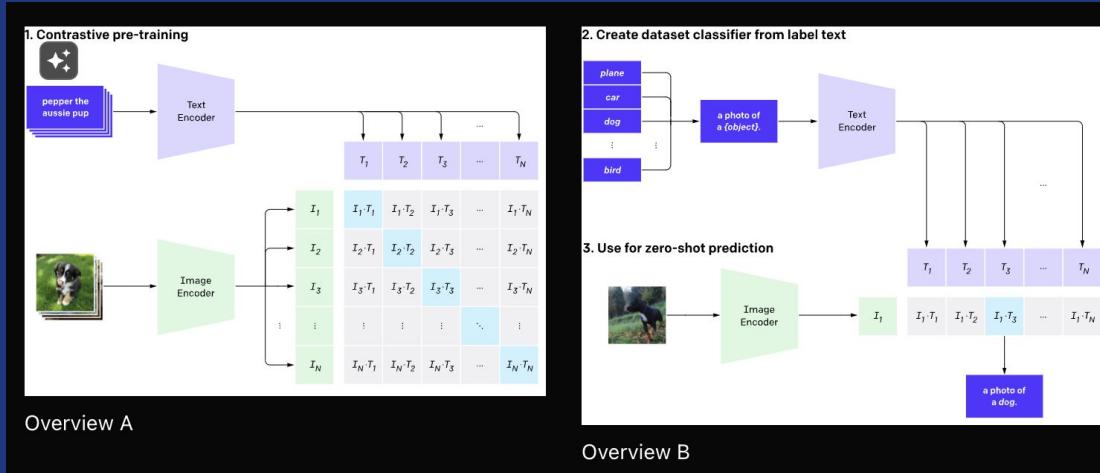
Como Funciona a Geração de Embeddings:

- Usamos o modelo **CLIP-ViT-B-32** para converter a imagem em um vetor de 512 dimensões.

```
dog = Dog('Luigi', 'path/para/luigi.png', 'Jack Russel', 'Ully')  
dog.generate_embedding()
```

(a imagem é convertida em um vetor usando o modelo de embeddings)

Anatomia de um modelo multimodal - CLIP da OpenAI



CLIP pre-trains an image encoder and a text encoder to predict which images were paired with which texts in our dataset. We then use this behavior to turn CLIP into a zero-shot classifier. We convert all of a dataset's classes into captions such as “a photo of a dog” and predict the class of the caption CLIP estimates best pairs with a given image.

```
from sentence_transformers import SentenceTransformer  
  
model = SentenceTransformer('clip-ViT-B-32')
```

Registrando Cães no Elasticsearch

- **Fluxo de Registro:**
 - Utilizamos a classe `DogRepository` para inserir um ou mais cães no índice do Elasticsearch.
- **Código de Registro:**

```
dog_repo = DogRepository(es, "dog-image-index")
dog_repo.insert(dog)
```

- **Diagrama de Registro de Dados:**
 - Inserção de múltiplos cães, cada um com sua imagem e dados no banco vetorial.

(Mostrar visualmente como várias imagens de cães são armazenadas e indexadas no Elasticsearch)

Case – Encontrando o Cão Perdido

Cenário:

- Uma pessoa encontra um cãozinho perdido e tira uma foto para tentar encontrar o dono.

Busca Vetorial no Elasticsearch:

- A foto do cão perdido é convertida em um embedding, e a busca por similaridade é feita no banco de dados vetorial.

```
result = dog_service.find_dog_by_image('path/para/lost_dog.png')  
(fluxo de busca: Foto -> Embedding -> Busca no Elasticsearch -> Retorno de Cães Semelhantes)
```

Testando a Busca por Imagens

- **Visualizando o Resultado:**

- A imagem do cão encontrado é comparada com as imagens no banco de dados, retornando a imagem e o nome do dono.

```
filename = result['hits'][‘hits’][0][‘_source’][‘image_path’]  
print(result[‘hits’][‘hits’][0][‘_source’][‘dog_id’])
```

-
- **Exemplo de Saída:**

- **Cão:** Luigi
- **Raça:** Jack Russel
- **Dono:** Ully

(Incluir visualização das imagens e resultados de similaridade da busca)

Prática – Buscando Outros Cães

- **Demonstração Prática:**
 - Os participantes podem testar o código com outras imagens de cães fornecidas no notebook.
 - **Atividade:** Fazer upload de uma nova imagem e verificar se a busca retorna os resultados corretos.

(Incluir instruções passo a passo para que os participantes possam testar por conta própria)

Resumo do tutorial prático

Resumo:

- Implementamos uma solução de busca de imagens usando bancos de dados vetoriais com Elasticsearch.
- Utilizamos o caso prático de encontrar o cão perdido **Luigi**, demonstrando como imagens podem ser armazenadas e buscadas eficientemente usando embeddings.

Próximos Passos:

- Escalar a solução para grandes volumes de imagens.
- Integrar com outras soluções de busca e detecção de objetos.

Recursos para desenvolvedores: Elasticsearch

elastic.co/search-labs

github.com/elastic/elasticsearch-labs

BLOG / ML RESEARCH

Evaluating RAG: A journey through metrics



In 2020, Meta published a paper titled "[Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks](#)". This paper introduced a method for expanding the knowledge of Language

A screenshot of a GitHub repository page for "elasticsearch-labs". The repository is public and contains guides, notebooks, and example apps for search applications. It includes sections for README, Apache-2.0 license, security policy, activity (109 stars, 178 watching, 40 forks), and a report repository. A language usage chart shows Jupyter Notebook as the primary language at 93.7%, followed by Python (2.9%), JavaScript (1.6%), TypeScript (1.3%), CSS (0.2%), Handlebars (0.1%), and Other (0.2%).

Language	Percentage
Jupyter Notebook	93.7%
Python	2.9%
JavaScript	1.6%
TypeScript	1.3%
CSS	0.2%
Handlebars	0.1%
Other	0.2%

**Generative AI
ML Research
Vector Search
How-Tos
Integrations
Lucene**

Recursos para desenvolvedores: Junte-se à Comunidade Elastic

Elastic User Groups

Estamos sempre em busca de organizadores, palestrantes e participantes.

Encontre mais eventos Elastic em todo o mundo em community.elastic.co



Meetup Elastic: Rio de Janeiro/RJ

elastic.co/community



Meetup Python Floripa: Florianópolis/SC



Gratidão

Alex Salgado
Developer Advocate @ Elastic

- | | |
|--|---|
|  @alexsalgadoprof |  @alexsalgadoprof |
|  salgado |  /in/alex-salgado/ |

