

Applying AI with Python

Alex Conway
alex@numberboost.com



Alex Conway

Founder NumberBoost, AiSupervision (YC W22)

Cape Town, Western Cape, South Africa · [Contact info](#)



NumberBoost



Y Combinator

x.com/alexconway
linkedin.com/in/alexconway/





Quantitative Analyst & Data Engineer

Mergence Investment Managers

Jul 2013 - Dec 2015 · 2 yrs 6 mos





Quantitative Analyst & Data Engineer

Mergence Investment Managers

Jul 2013 - Dec 2015 · 2 yrs 6 mos



Data Scientist

Superbalist.com

Feb 2016 - Jan 2017 · 1 yr





Quantitative Analyst & Data Engineer

Mergence Investment Managers

Jul 2013 - Dec 2015 · 2 yrs 6 mos



Data Scientist

Superbalist.com

Feb 2016 - Jan 2017 · 1 yr



Founder, CEO

NumberBoost

Jun 2016 - Present · 8 yrs 4 mos

We are multi-award winning AI developers.

- * Won Mercedes-Benz Innovation Competition
- * Won NTT Japan Open Innovation Challenge ZA
- * Won Lloyd's Register London A.I. Real-time Error Detection Challenge
- * Won MultiChoice Innovation Competition
- * Won Best on-chain AI agent(s) built on Galadriel
- * Won Best Use of CoopHive for AI Applications





Quantitative Analyst & Data Engineer

Mergence Investment Managers

Jul 2013 - Dec 2015 · 2 yrs 6 mos



Data Scientist

Superbalist.com

Feb 2016 - Jan 2017 · 1 yr



Founder, CEO

NumberBoost

Jun 2016 - Present · 8 yrs 4 mos

We are multi-award winning AI developers.

- * Won Mercedes-Benz Innovation Competition
- * Won NTT Japan Open Innovation Challenge ZA
- * Won Lloyd's Register London A.I. Real-time Error Detection Challenge
- * Won MultiChoice Innovation Competition
- * Won Best on-chain AI agent(s) built on Galadriel
- * Won Best Use of CoopHive for AI Applications



Co-founder, CTO

AiSupervision (YC W22)

Sep 2020 - May 2023 · 2 yrs 9 mos

The operating system for factory production lines.





Quantitative Analyst & Data Engineer

Mergence Investment Managers

Jul 2013 - Dec 2015 · 2 yrs 6 mos



Data Scientist

Superbalist.com

Feb 2016 - Jan 2017 · 1 yr



Founder, CEO

NumberBoost

Jun 2016 - Present · 8 yrs 4 mos

We are multi-award winning AI developers.

- * Won Mercedes-Benz Innovation Competition
- * Won NTT Japan Open Innovation Challenge ZA
- * Won Lloyd's Register London A.I. Real-time Error Detection Challenge
- * Won MultiChoice Innovation Competition
- * Won Best on-chain AI agent(s) built on Galadriel
- * Won Best Use of CoopHive for AI Applications



Co-founder, CTO

AiSupervision (YC W22)

Sep 2020 - May 2023 · 2 yrs 9 mos

The operating system for factory production lines.



Y Combinator

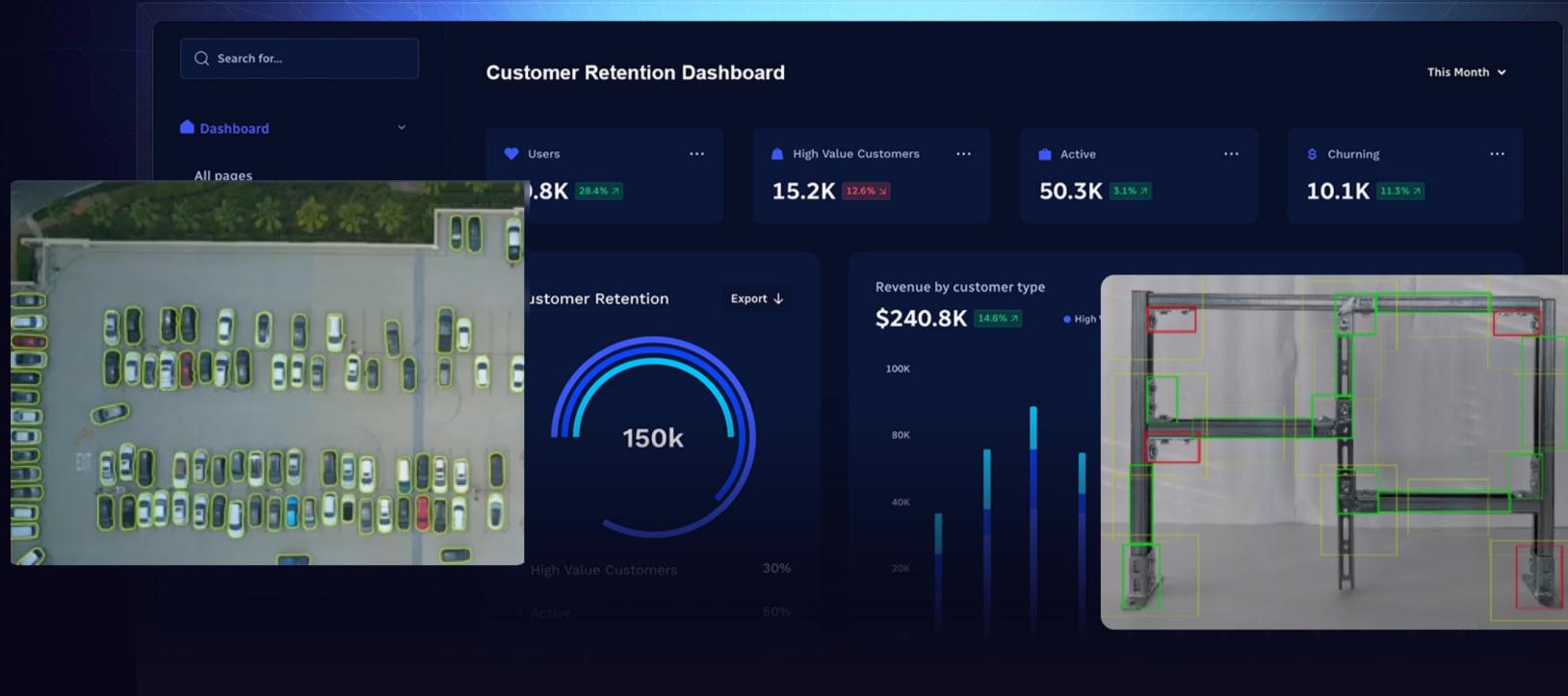
W22



WE JUST CELEBRATED OUR 8 YEAR ANNIVERSARY!

We develop custom AI solutions that 10x efficiency

We build artificial intelligence and data science systems that measurably improve business outcomes. Fast.

[Start Your AI Project](#)

jupyter Agent Last Checkpoint: 16 hours ago

File Edit View Run Kernel Settings Help Trusted

Define Agent Pipeline

```
[118...]def agent(question, max_retries=2):
    attempt = 0
    answers_question = "no"
    while answers_question.lower() != "yes" and attempt < max_retries:
        attempt += 1
        if attempt > 1:
            print(f"> Rewriting question {attempt}: {question}")
            clear_notebook_output()
        # figure out if SQL or LLM question
        prompt Decide_LLM_or_SQL = get_prompt_Decide_LLM_or_SQL(question)
        question_type = run_chat_query(prompt Decide_LLM_or_SQL)
        # reasoning
        if question_type == "SQL":
```

8 notebooks
150 slides
let's gooooo



[main](#)[1 Branch](#)[0 Tags](#)[Go to file](#)[Add file](#)[Code](#)

alxcnwy update readme

c4738c7 · 1 hour ago

23 Commits



code

update readme

1 hour ago



images

update readme

1 hour ago



.gitignore

clean up

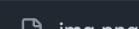
5 hours ago



README.md

readme

5 hours ago



img.png

readme

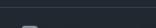
5 hours ago



requirements.txt

add text scraping

yesterday



slides.pdf

update readme

5 hours ago



slides.pptx

update readme

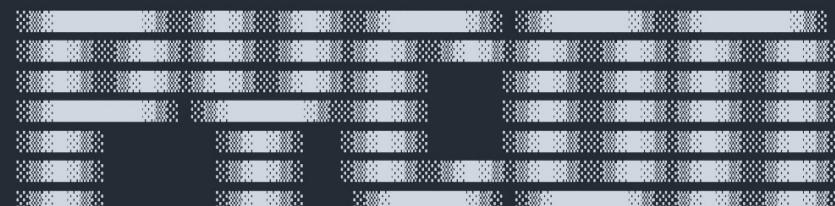
1 hour ago



README



Applying AI with Python



Slides and code for ["Applying AI with Python"](#) talk at PyConZA 2024.

This code introduces a novel agent architecture that demonstrates self-reflection, tool use and reasoning.

About



Slides and Code for my talk at PyConZA 2024 on "Applying AI with Python"

Readme

Activity

1 star

1 watching

0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Languages

• Jupyter Notebook 100.0%





github.com/alexchnwy/applying-ai-with-python



What is Artificial Intelligence (AI)?



what is artificial intelligence



Artificial Intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to perform tasks that typically require human intelligence. At the core of AI are models that take input data, apply transformations to it using learned patterns or algorithms, and produce a desired output.





what is artificial intelligence



Artificial Intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to perform tasks that typically require human intelligence. At the core of AI are models that take input data, apply transformations to it using learned patterns or algorithms, and produce a desired output.



```
def ai_model(input):  
    #  
    # algorithms f(input) = output  
    #  
    return output
```



```
def ai_model(input):  
    #  
    # algorithms f(input) = output  
    #  
    return output
```



	A	B	C	D
1	Experience_Years	Age	Postgrad	Salary
2		5	28	FALSE
3		1	21	TRUE
4		3	23	FALSE
5		2	22	TRUE
6		1	17	TRUE
7		25	62	TRUE
8		19	54	FALSE
9		2	21	FALSE
10		10	36	FALSE
				61500

Table

Artificial Intelligence (AI) is transforming industries and our lives by enabling computers to solve increasingly complex tasks, and Python stands at the forefront of this revolution. In this talk, we will explore how Python's rich ecosystem of AI libraries and frameworks can be harnessed to

Text

Input



Image / Video



Audio



```
def ai_model(input):  
    #  
    # algorithms f(input) = output  
    #  
    return output
```

Algorithm Terminology

- **(Supervised) Regression** (predict a number)
- **(Supervised) Classification** (predict a category / label)
- **Unsupervised** (find patterns within a dataset without labels)
- **Generative** (create new data / content)

Types of AI Algorithms

- **Regression** (Linear / Polynomial / Logistic, etc.)
- **Decision Trees** (Random Forests, GBMs, etc.)
- **Clustering** (K-Means, Spectral, DBSCAN, etc.)
- **Neural Networks** (Convolutional, Recurrent, Transformer, etc.)
- etc...

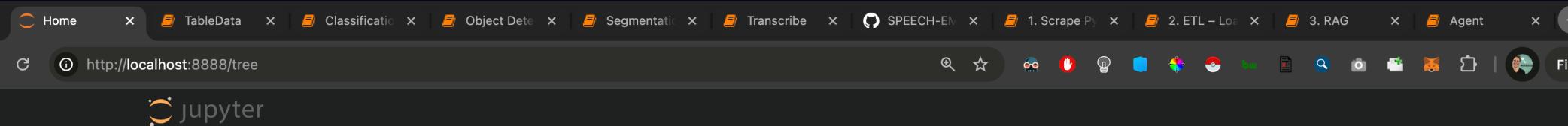


Don't Panic!

No Math Here

can you copy & paste?

```
def ai_model(input):  
    # Examples incoming (brace yo'self)  
    #  
    #  
    return output
```



```
def ai_model(input):  
    #  
    # algorithms f(input) = output  
    #  
    return output
```



	A	B	C	D
1	Experience_Years	Age	Postgrad	Salary
2		5	28	FALSE
3		1	21	TRUE
4		3	23	FALSE
5		2	22	TRUE
6		1	17	TRUE
7		25	62	TRUE
8		19	54	FALSE
9		2	21	FALSE
10		10	36	FALSE
				61500

Table

Artificial Intelligence (AI) is transforming industries and our lives by enabling computers to solve increasingly complex tasks, and Python stands at the forefront of this revolution. In this talk, we will explore how Python's rich ecosystem of AI libraries and frameworks can be harnessed to

Text

Output



Image / Video



Audio



	A	B	C	D
1	Experience_Years	Age	Postgrad	Salary
2		5	28	FALSE
3		1	21	TRUE
4		3	23	FALSE
5		2	22	TRUE
6		1	17	TRUE
7		25	62	TRUE
8		19	54	FALSE
9		2	21	FALSE
10		10	36	FALSE
				61500

Table

Artificial Intelligence (AI) is transforming industries and our lives by enabling computers to solve increasingly complex tasks, and Python stands at the forefront of this revolution. In this talk, we will explore how Python's rich ecosystem of AI libraries and frameworks can be harnessed to

Text

Output

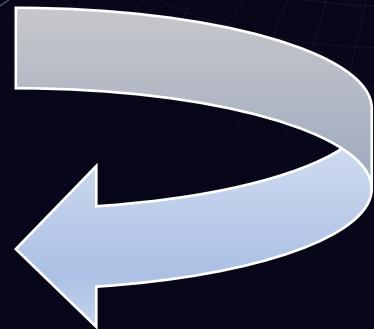


Image / Video



Audio



	A	B	C	D
1	Experience_Years	Age	Postgrad	Salary
2	5	28	FALSE	250000
3	1	21	TRUE	50000
4	3	23	FALSE	170000
5	2	22	TRUE	25000
6	1	17	TRUE	10000
7	25	62	TRUE	5001000
8	19	54	FALSE	800000
9	2	21	FALSE	9000
10	10	36	FALSE	61500

Table

Artificial Intelligence (AI) is transforming industries and our lives by enabling computers to solve increasingly complex tasks, and Python stands at the forefront of this revolution. In this talk, we will explore how Python's rich ecosystem of AI libraries and frameworks can be harnessed to

Text

Output



Image / Video



Audio



	A	B	C	D
1	Experience_Years	Age	Postgrad	Salary
2	5	28	FALSE	250000
3	1	21	TRUE	50000
4	3	23	FALSE	170000
5	2	22	TRUE	25000
6	1	17	TRUE	10000
7	25	62	TRUE	5001000
8	19	54	FALSE	800000
9	2	21	FALSE	9000
10	10	36	FALSE	61500

Table

Artificial Intelligence (AI) is transforming industries and our lives by enabling computers to solve increasingly complex tasks, and Python stands at the forefront of this revolution. In this talk, we will explore how Python's rich ecosystem of AI libraries and frameworks can be harnessed to

Text

Output



Image / Video



Audio



	A	B	C	D
1	Experience_Years	Age	Postgrad	Salary
2	5	28	FALSE	250000
3	1	21	TRUE	50000
4	3	23	FALSE	170000
5	2	22	TRUE	25000
6	1	17	TRUE	10000
7	25	62	TRUE	5001000
8	19	54	FALSE	800000
9	2	21	FALSE	9000
10	10	36	FALSE	61500

Table

Artificial Intelligence (AI) is transforming industries and our lives by enabling computers to solve increasingly complex tasks, and Python stands at the forefront of this revolution. In this talk, we will explore how Python's rich ecosystem of AI libraries and frameworks can be harnessed to

Text

Output



Image / Video



Audio



	A	B	C	D
1	Experience_Years	Age	Postgrad	Salary
2		5	28	FALSE
3		1	21	TRUE
4		3	23	FALSE
5		2	22	TRUE
6		1	17	TRUE
7		25	62	TRUE
8		19	54	FALSE
9		2	21	FALSE
10		10	36	FALSE
				61500

Table

Artificial Intelligence (AI) is transforming industries and our lives by enabling computers to solve increasingly complex tasks, and Python stands at the forefront of this revolution. In this talk, we will explore how Python's rich ecosystem of AI libraries and frameworks can be harnessed to

Text

Output

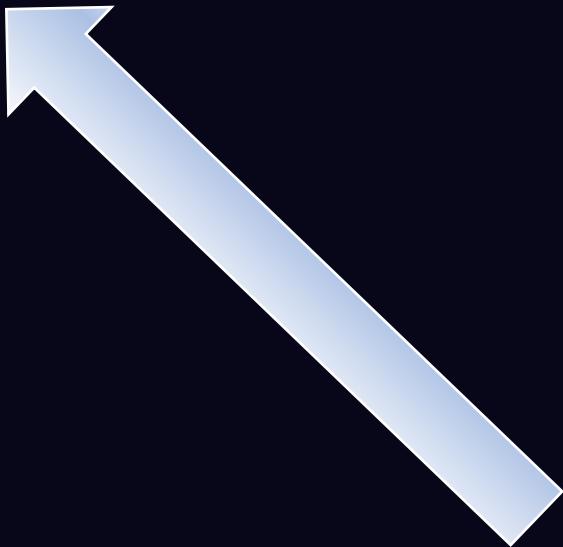


Image / Video



Audio



	A	B	C	D
1	Experience_Years	Age	Postgrad	Salary
2	5	28	FALSE	250000
3	1	21	TRUE	50000
4	3	23	FALSE	170000
5	2	22	TRUE	25000
6	1	17	TRUE	10000
7	25	62	TRUE	5001000
8	19	54	FALSE	800000
9	2	21	FALSE	9000
10	10	36	FALSE	61500

Table

Artificial Intelligence (AI) is transforming industries and our lives by enabling computers to solve increasingly complex tasks, and Python stands at the forefront of this revolution. In this talk, we will explore how Python's rich ecosystem of AI libraries and frameworks can be harnessed to

Text

Output



Image / Video



Audio



	A	B	C	D
1	Experience_Years	Age	Postgrad	Salary
2		5	28	FALSE
3		1	21	TRUE
4		3	23	FALSE
5		2	22	TRUE
6		1	17	TRUE
7		25	62	TRUE
8		19	54	FALSE
9		2	21	FALSE
10		10	36	FALSE
				61500

Table

Artificial Intelligence (AI) is transforming industries and our lives by enabling computers to solve increasingly complex tasks, and Python stands at the forefront of this revolution. In this talk, we will explore how Python's rich ecosystem of AI libraries and frameworks can be harnessed to

Text

Output



Image / Video



Audio

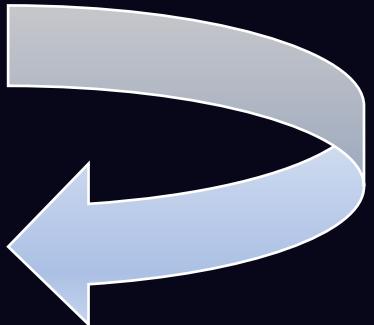


	A	B	C	D
1	Experience_Years	Age	Postgrad	Salary
2		5	28	FALSE
3		1	21	TRUE
4		3	23	FALSE
5		2	22	TRUE
6		1	17	TRUE
7		25	62	TRUE
8		19	54	FALSE
9		2	21	FALSE
10		10	36	FALSE
				61500

Table

Artificial Intelligence (AI) is transforming industries and our lives by enabling computers to solve increasingly complex tasks, and Python stands at the forefront of this revolution. In this talk, we will explore how Python's rich ecosystem of AI libraries and frameworks can be harnessed to

Text



Output



Image / Video



Audio



	A	B	C	D
1	Experience_Years	Age	Postgrad	Salary
2		5	28	FALSE
3		1	21	TRUE
4		3	23	FALSE
5		2	22	TRUE
6		1	17	TRUE
7		25	62	TRUE
8		19	54	FALSE
9		2	21	FALSE
10		10	36	FALSE
				61500

Table

Artificial Intelligence (AI) is transforming industries and our lives by enabling computers to solve increasingly complex tasks, and Python stands at the forefront of this revolution. In this talk, we will explore how Python's rich ecosystem of AI libraries and frameworks can be harnessed to

Text

Output

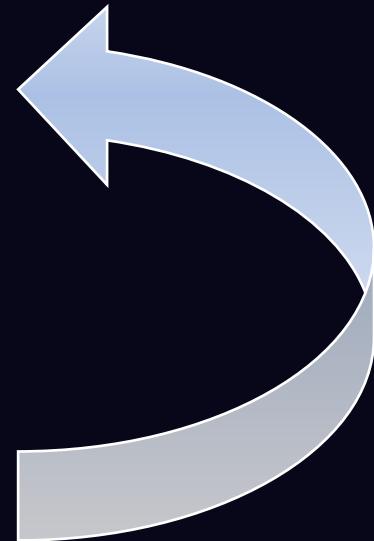


Image / Video



Audio



Why Python for AI?

- Easy to use (ML researchers often != software engineers)
- Massive community (easy to find answers to questions)
- Extensive libraries and frameworks (don't need to invent to use)

 TensorFlow



 PyTorch



 Keras

Hugging Face

 XGBoost

 statsmodels

 pandas





Search models, datasets, users...

Models Datasets Spaces Posts Docs Solutions Pricing Log In Sign Up

Tasks Libraries Datasets Languages Licenses Other

Filter Tasks by name

Multimodal

Computer Vision

Natural Language Processing

Audio

Tabular

Reinforcement Learning

Models 1,026,955 Filter by name

Full-text search

↑↓ Sort: Trending

- [meta-llama/Llama-3.2-11B-Vision-Instruct](#)
Image-Text-to-Text • Updated 3 days ago • ↓ 122k • ⚡ • ❤ 469
- [openai/whisper-large-v3-turbo](#)
Automatic Speech Recognition • Updated about 22 hours ago • ↓ 5.73k • ❤ 263
- [black-forest-labs/FLUX.1-dev](#)
Text-to-Image • Updated Aug 16 • ↓ 1.12M • ⚡ • ❤ 5.02k
- [jasperai/Flux.1-dev-Controlnet-Upscaler](#)
Image-to-Image • Updated 3 days ago • ↓ 7.86k • ❤ 235
- [nvidia/NVLM-D-72B](#)
Image-Text-to-Text • Updated about 8 hours ago • ↓ 193 • ❤ 206
- [meta-llama/Llama-3.2-1B-Instruct](#)
Text Generation • Updated 8 days ago • ↓ 88.8k • ❤ 267
- [allenai/Molmo-7B-D-0924](#)
Image-Text-to-Text • Updated 1 day ago • ↓ 13k • ❤ 267
- [stepfun-ai/GOT-OCR2_0](#)
Image-Text-to-Text • Updated 15 days ago • ↓ 160k • ❤ 901
- [openbmb/MiniCPM-Embedding](#)
Feature Extraction • Updated 1 day ago • ↓ 116k • ❤ 199
- [meta-llama/Llama-3.2-3B-Instruct](#)
Text Generation • Updated 8 days ago • ↓ 69.7k • ❤ 241
- [allenai/Molmo-72B-0924](#)
Image-Text-to-Text • Updated 1 day ago • ↓ 1.63k • ❤ 196
- [meta-llama/Llama-3.2-11B-Vision](#)
Image-Text-to-Text • Updated 6 days ago • ↓ 37.1k • ❤ 206
- [BAAI/Emu3-Gen](#)
Text Generation • Updated 4 days ago • ↓ 1.25k • ❤ 94
- [OnomaAIResearch/Illustrious-xl-early-release-v0](#)
Text-to-Image • Updated 2 days ago • ↓ 3.56k • ⚡ • ❤ 160
- [meta-llama/Llama-3.2-90B-Vision-Instruct](#)
Image-Text-to-Text • Updated 3 days ago • ↓ 11.9k • ❤ 162
- [amd/AMD-Llama-135m](#)
Text Generation • Updated Aug 6 • ↓ 2.77k • ❤ 89
- [ostris/OpenFLUX.1](#)
Text-to-Image • Updated 2 days ago • ↓ 142 • ❤ 167
- [allenai/Molmo-7B-0-0924](#)
Image-Text-to-Text • Updated 1 day ago • ↓ 1.96k • ❤ 112
- [vidore/colqwen2-v0.1](#)
Updated 1 day ago • ↓ 3.01k • ❤ 70
- [meta-llama/Llama-3.1-8B-Instruct](#)
Text Generation • Updated 8 days ago • ↓ 3.23M • ⚡ • ❤ 2.69k
- [black-forest-labs/FLUX.1-schnell](#)
Text-to-Image • Updated Aug 16 • ↓ 939k • ⚡ • ❤ 2.44k
- [nvidia/Llama-3_1-Nemotron-51B-Instruct](#)
Text Generation • Updated 3 days ago • ↓ 13.6k • ❤ 164
- [glif/how2draw](#)
Text-to-Image • Updated 10 days ago • ↓ 4.75k • ⚡ • ❤ 141
- [openai/whisper-large-v3](#)
Automatic Speech Recognition • Updated Aug 12 • ↓ 4.14M • ⚡ • ❤ 3.5k
- [allenai/MolmoE-1B-0924](#)
Image-Text-to-Text • Updated 2 days ago • ↓ 4.64k • ❤ 85
- [AIDC-AI/Ovis1.6-Gemma2-9B](#)
Image-Text-to-Text • Updated 7 days ago • ↓ 6.58k • ❤ 184





Text

Audio

Table

Computer Vision

- Depth Estimation
- Image Classification
- Object Detection
- Image Segmentation
- Text-to-Image
- Image-to-Text
- Image-to-Image
- Image-to-Video
- Unconditional Image Generation
- Video Classification
- Text-to-Video
- Zero-Shot Image Classification
- Mask Generation
- Zero-Shot Object Detection
- Text-to-3D
- Image-to-3D
- Image Feature Extraction
- Keypoint Detection

Natural Language Processing

- Text Classification
- Token Classification
- Table Question Answering
- Question Answering
- Zero-Shot Classification
- Translation
- Summarization
- Feature Extraction
- Text Generation
- Text2Text Generation
- Fill-Mask
- Sentence Similarity

Audio

- Text-to-Speech
- Text-to-Audio
- Automatic Speech Recognition
- Audio-to-Audio
- Audio Classification
- Voice Activity Detection

Tabular

- Tabular Classification
- Tabular Regression
- Time Series Forecasting

Table Data



Wine Quality

Donated on 10/6/2009

Two datasets are included, related to red and white vinho verde wine samples, from the north of Portugal. The goal is to model wine quality based on physicochemical tests (see [Cortez et al., 2009], <http://www3.dsi.uminho.pt/pcortez/wine/>).

Dataset Characteristics

Multivariate

Subject Area

Business

Associated Tasks

Classification, Regression

Feature Type

Real

Instances

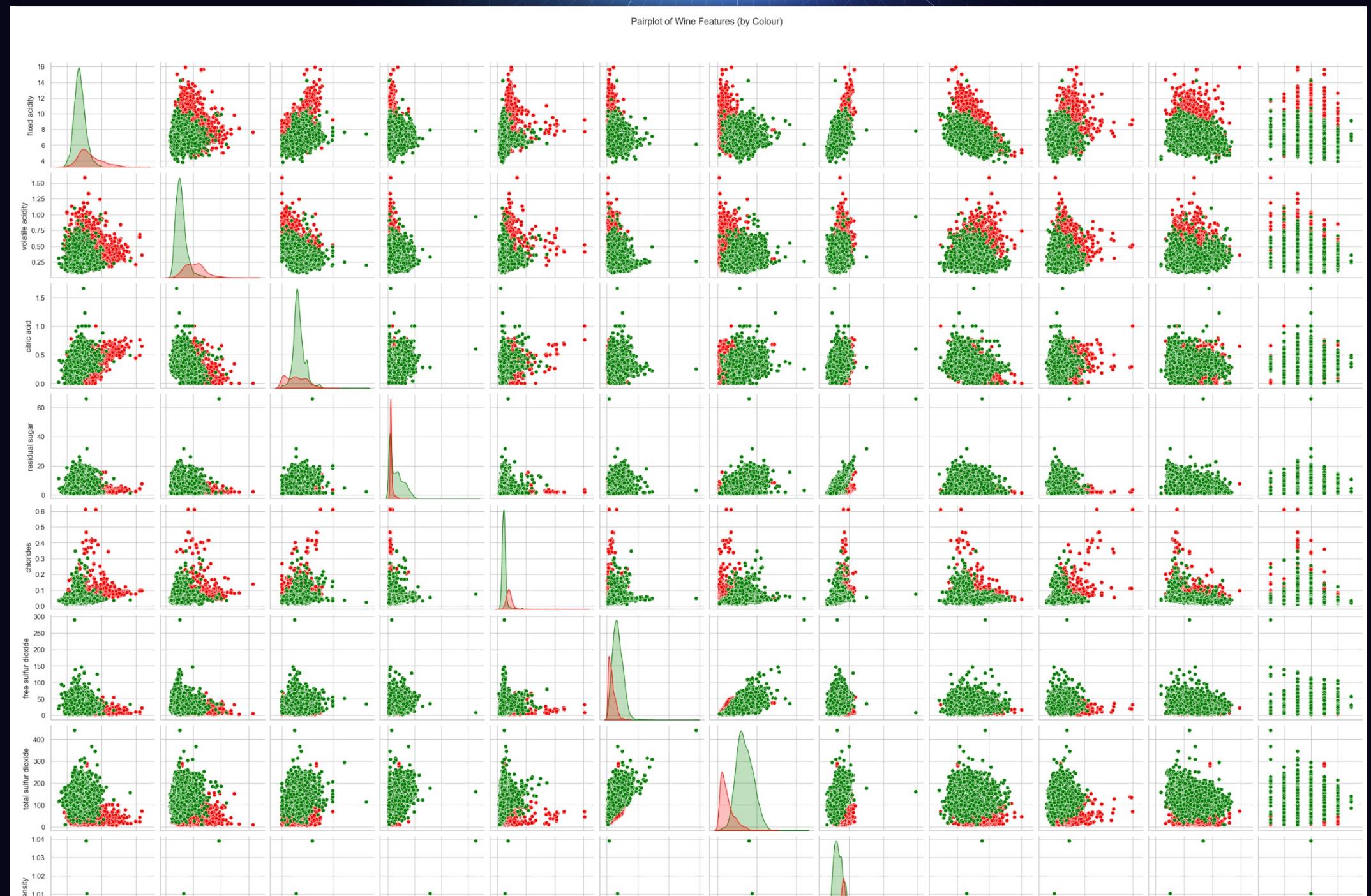
4898

Features

11

<https://archive.ics.uci.edu/dataset/186/wine+quality>





	A	B	C	D	E	F	G	H	I	J	K	L	M
1	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	color	quality
2	7.4	0.7	0	1.9	0.076	11	34	0.9978	3.51	0.56	9.4	0	5
3	7.8	0.88	0	2.6	0.098	25	67	0.9968	3.2	0.68	9.8	0	5
4	7.8	0.76	0.04	2.3	0.092	15	54	0.997	3.26	0.65	9.8	0	5
5	11.2	0.28	0.56	1.9	0.075	17	60	0.998	3.16	0.58	9.8	0	6
6	7.4	0.7	0	1.9	0.076	11	34	0.9978	3.51	0.56	9.4	0	5
7	7.4	0.66	0	1.8	0.075	13	40	0.9978	3.51	0.56	9.4	0	5
8	7.9	0.6	0.06	1.6	0.069	15	59	0.9964	3.3	0.46	9.4	0	5
9	7.3	0.65	0	1.2	0.065	15	21	0.9946	3.39	0.47	10	0	7
10	7.8	0.58	0.02	2	0.073	9	18	0.9968	3.36	0.57	9.5	0	7
11	7.5	0.5	0.36	6.1	0.071	17	102	0.9978	3.35	0.8	10.5	0	5
12	6.7	0.58	0.08	1.8	0.097	15	65	0.9959	3.28	0.54	9.2	0	5
13	7.5	0.5	0.36	6.1	0.071	17	102	0.9978	3.35	0.8	10.5	0	5
14	5.6	0.615	0	1.6	0.089	16	59	0.9943	3.58	0.52	9.9	0	5
15	7.8	0.61	0.29	1.6	0.114	9	29	0.9974	3.26	1.56	9.1	0	5
16	8.9	0.62	0.18	3.8	0.176	52	145	0.9986	3.16	0.88	9.2	0	5
17	8.9	0.62	0.19	3.9	0.17	51	148	0.9986	3.17	0.93	9.2	0	5
18	8.5	0.28	0.56	1.8	0.092	35	103	0.9969	3.3	0.75	10.5	0	7
19	8.1	0.56	0.28	1.7	0.368	16	56	0.9968	3.11	1.28	9.3	0	5
20	7.4	0.59	0.08	4.4	0.086	6	29	0.9974	3.38	0.5	9	0	4
21	7.9	0.32	0.51	1.8	0.341	17	56	0.9969	3.04	1.08	9.2	0	6
22	8.9	0.22	0.48	1.8	0.077	29	60	0.9968	3.39	0.53	9.4	0	6
23	7.6	0.39	0.31	2.3	0.082	23	71	0.9982	3.52	0.65	9.7	0	5
24	7.9	0.43	0.21	1.6	0.106	10	37	0.9966	3.17	0.91	9.5	0	5
25	8.5	0.49	0.11	2.3	0.084	9	67	0.9968	3.17	0.53	9.4	0	5
26	6.9	0.4	0.14	2.4	0.085	21	40	0.9968	3.43	0.63	9.7	0	6

Regression Model to Predict Wine Quality



Jupyter Notebook

TableData.ipynb



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	color	quality
2	7.4	0.7	0	1.9	0.076	11	34	0.9978	3.51	0.56	9.4	red	5
3	7.8	0.88	0	2.6	0.098	25	67	0.9968	3.2	0.68	9.8	red	5
4	7.8	0.76	0.04	2.3	0.092	15	54	0.997	3.26	0.65	9.8	red	5
5	11.2	0.28	0.56	1.9	0.075	17	60	0.998	3.16	0.58	9.8	red	6
6	7.4	0.7	0	1.9	0.076	11	34	0.9978	3.51	0.56	9.4	red	5
7	7.4	0.66	0	1.8	0.075	13	40	0.9978	3.51	0.56	9.4	red	5
8	7.9	0.6	0.06	1.6	0.069	15	59	0.9964	3.3	0.46	9.4	red	5
9	7.3	0.65	0	1.2	0.065	15	21	0.9946	3.39	0.47	10	red	7
10	7.8	0.58	0.02	2	0.073	9	18	0.9968	3.36	0.57	9.5	red	7
11	7.5	0.5	0.36	6.1	0.071	17	102	0.9978	3.35	0.8	10.5	red	5
12	6.7	0.58	0.08	1.8	0.097	15	65	0.9959	3.28	0.54	9.2	red	5
13	7.5	0.5	0.36	6.1	0.071	17	102	0.9978	3.35	0.8	10.5	red	5
14	5.6	0.615	0	1.6	0.089	16	59	0.9943	3.58	0.52	9.9	red	5
15	7.8	0.61	0.29	1.6	0.114	9	29	0.9974	3.26	1.56	9.1	red	5
16	8.9	0.62	0.18	3.8	0.176	52	145	0.9986	3.16	0.88	9.2	red	5
17	8.9	0.62	0.19	3.9	0.17	51	148	0.9986	3.17	0.93	9.2	red	5
18	8.5	0.28	0.56	1.8	0.092	35	103	0.9969	3.3	0.75	10.5	red	7
19	8.1	0.56	0.28	1.7	0.368	16	56	0.9968	3.11	1.28	9.3	red	5
20	7.4	0.59	0.08	4.4	0.086	6	29	0.9974	3.38	0.5	9	red	4
21	7.9	0.32	0.51	1.8	0.341	17	56	0.9969	3.04	1.08	9.2	red	6
22	8.9	0.22	0.48	1.8	0.077	29	60	0.9968	3.39	0.53	9.4	red	6
23	7.6	0.39	0.31	2.3	0.082	23	71	0.9982	3.52	0.65	9.7	red	5
24	7.9	0.43	0.21	1.6	0.106	10	37	0.9966	3.17	0.91	9.5	red	5
25	8.5	0.49	0.11	2.3	0.084	9	67	0.9968	3.17	0.53	9.4	red	5
26	6.9	0.4	0.14	2.4	0.085	21	40	0.9968	3.43	0.63	9.7	red	6

Classification Model to Predict Wine Colour



Jupyter Notebook

TableData.ipynb



	A	B	C	D
1	Recency	Frequency	Monetary Value	Churned
2	13	3	983	0
3	62	7	1090	0
4	29	3	292	0
5	10	5	226	1
6	176	7	802	1
7	48	5	252	0
8	137	9	1825	1
9	177	3	1888	0
10	64	9	985	1
11	175	5	182	0
12	47	7	1545	1
13	110	9	378	1
14	178	10	1632	0
15	125	4	1824	1





Hugging Face

Tabular



Tabular Classification



Tabular Regression



Time Series Forecasting





aws amazon/chronos-t5-tiny

like 86

Time Series Forecasting Transformers Safetensors t5 text2text-generation time series forecasting pretrained models foundation models time series foundation models time-series
text-generation-inference Inference Endpoints arxiv:2403.07815 arxiv:1910.10683 License: apache-2.0

Model card

Files and versions

Community 4

⋮

Train

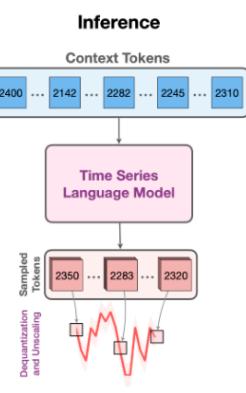
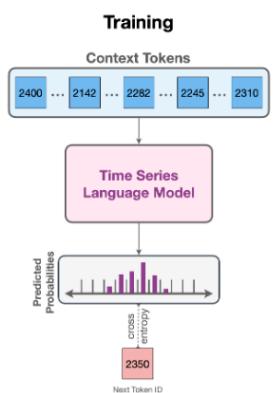
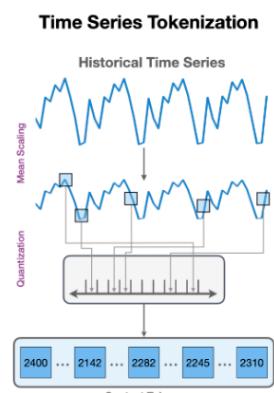
Deploy

Use this model

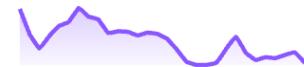
Chronos-T5 (Tiny)

Chronos is a family of **pretrained time series forecasting models** based on language model architectures. A time series is transformed into a sequence of tokens via scaling and quantization, and a language model is trained on these tokens using the cross-entropy loss. Once trained, probabilistic forecasts are obtained by sampling multiple future trajectories given the historical context. Chronos models have been trained on a large corpus of publicly available time series data, as well as synthetic data generated using Gaussian processes.

For details on Chronos models, training data and procedures, and experimental results, please refer to the paper [Chronos: Learning the Language of Time Series](#).



Downloads last month
14,029,406



Safetensors

Model size 8.39M params Tensor type F32

Inference API

Time Series Forecasting

Inference API (serverless) does not yet support transformers models for this pipeline type.

Collection including amazon/chronos-t5-tiny

Chronos Models & Datasets

Chronos: Pretrained (language) models fo... • 8 items • Updated Jun 27 • △ 28





aws amazon/chronos-t5-tiny

How to use from the **Transformers** library

```
# Load model directly
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM

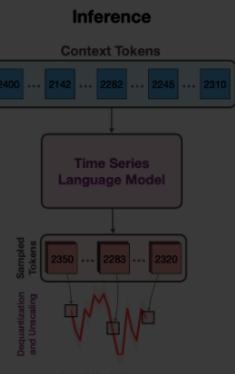
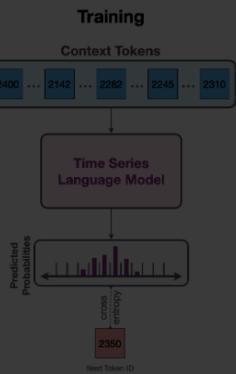
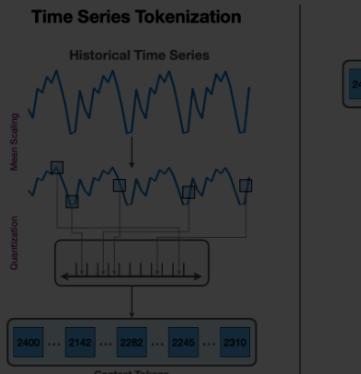
tokenizer = AutoTokenizer.from_pretrained("amazon/chronos-t5-tiny")
model = AutoModelForSeq2SeqLM.from_pretrained("amazon/chronos-t5-tiny")
```

 Copy[Model card](#) [Files and versions](#)

Chronos-T5 (Tiny)

Chronos is a family of **pretrained time series** architectures. A time series is transformed into tokens and a language model is trained on these tokens. Probabilistic forecasts are obtained by sampling context. Chronos models have been trained on a large corpus of publicly available time series data, as well as synthetic data generated using Gaussian processes.

For details on Chronos models, training data and procedures, and experimental results, please refer to the paper [Chronos: Learning the Language of Time Series](#).

[odels](#) [time-series](#)[Train](#) [Deploy](#) [Use this model](#)

odel size 8.39M params Tensor type F32

[Time Series Forecasting](#)

Inference API (serverless) does not yet support transformers models for this pipeline type.

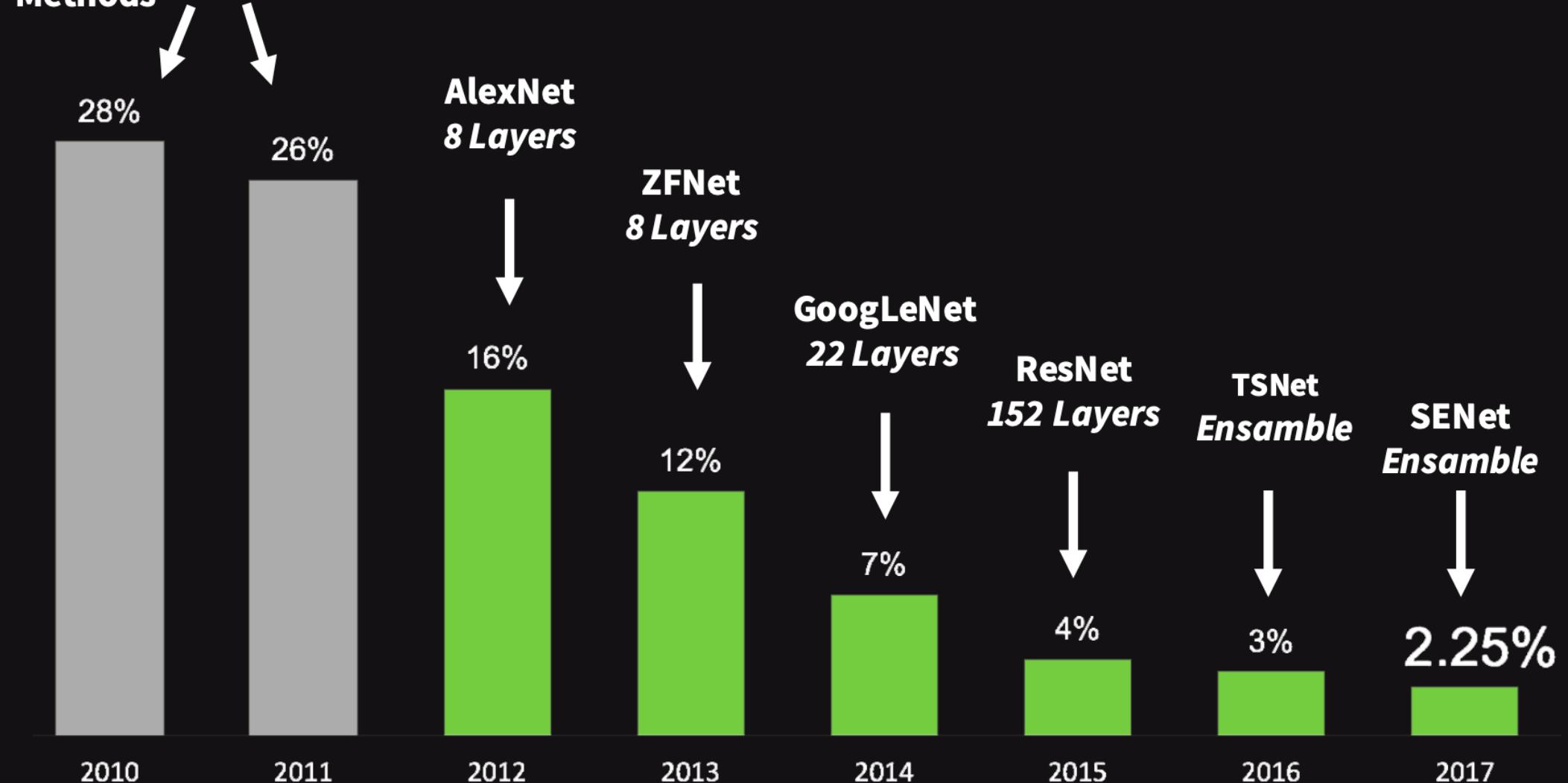
[Collection including amazon/chronos-t5-tiny](#)[Chronos Models & Datasets](#) [Collection](#)

Chronos: Pretrained (language) models fo... • 8 items • Updated Jun 27 • △ 28



Image / Video Data

**Traditional
Image Processing
Methods**



Human error = ~ 5%



Dog, domestic dog, *Canis familiaris*

A member of the genus *Canis* (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds; "the dog barked all night"

1603
pictures

88.15%
Popularity
Percentile

World
IDs

Treemap Visualization

Images of the Synset

Downloads

[Home](#) > [ImageNet 2011 Fall Release](#) > [A](#) > [Domestic animal, domesticated animal](#) > [Dog, domestic dog, *Canis familiaris*](#)

Hunting



Working



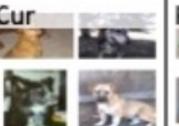
Toy



Great



Cur



Pug



Dalmatian



Basenji



Mexican



Poodle



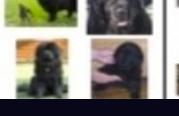
Lapdog



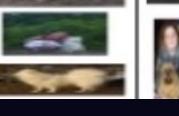
Leonberg



Puppy



Newfoundlan

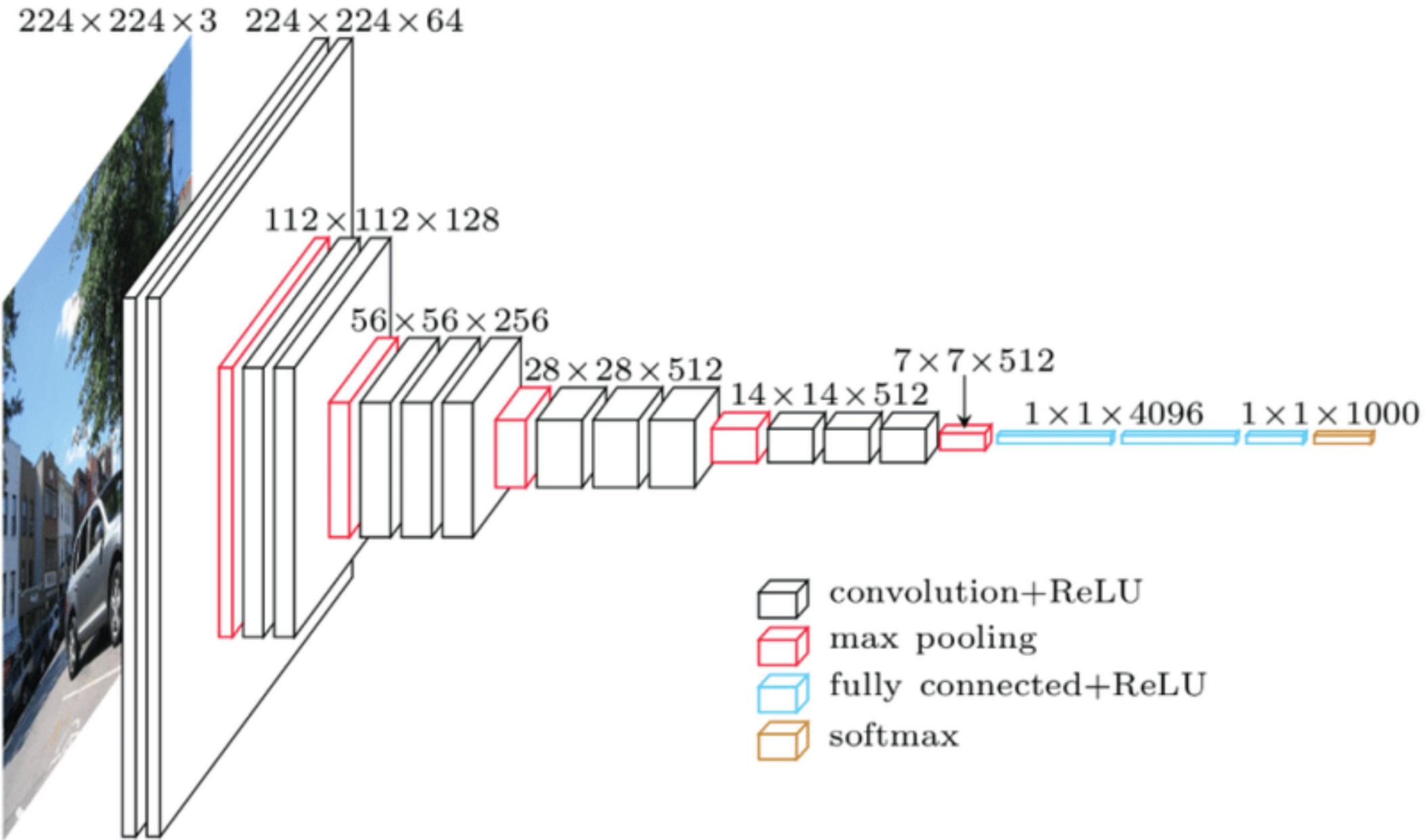


Spitz



- animal, animate being, beast, brute, creature, fauna
 - invertebrate (766)
 - homeotherm, homoiotherm,恒温动物
 - work animal (4)
 - darter (0)
 - survivor (0)
 - range animal (0)
 - creepy-crawly (0)
- domestic animal, domesticated animal
 - domestic cat, house cat, Feline, Felidae (4)
 - dog, domestic dog, *Canis familiaris* (1603)
 - pooch, doggie, doggy, dog (101)
 - sporting dog, gun dog (1)
 - dachshund, dachsie, sausage dog, sau (1)
 - terrier (37)
 - coursing (0)
 - hound, hound dog (1)
 - Rhodesian ridgeback (0)
 - dalmatian, coach dog, carriage dog (0)
 - cur, mongrel, mutt (2)
 - corgi, Welsh corgi (2)
 - Mexican hairless (0)
 - lapdog (0)
 - Newfoundland, Newfou (0)
 - poodle, poodle dog (4)
 - basenji (0)
 - Leonberg (0)
 - griffon, Brussels griffor (0)
 - pug, pug-dog (0)
 - working dog (45)





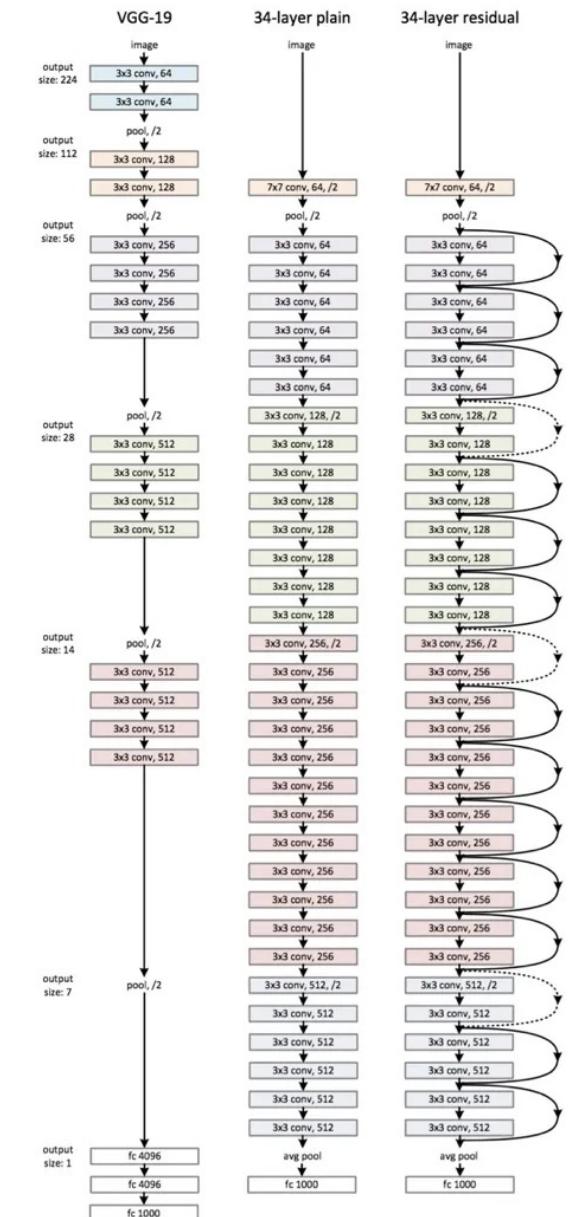


Figure 3. Example network architectures for ImageNet. **Left:** the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle:** a plain network with 34 parameter layers (3.6 billion FLOPs). **Right:** a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.



Jupyter Notebook

Let's use a pre-trained convolutional neural network!

Classification and Visual Search.ipynb



```
# Load the ResNet50 model with pretrained weights (ImageNet)
model = ResNet50(weights='imagenet')

def run_pretrained_classifier(img_path):
    img = load_img(img_path, target_size=(224, 224))
    img_array = img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0)
    img_array = preprocess_input(img_array)
    predictions = model.predict(img_array)
    return tf.keras.applications.resnet50.decode_predictions(predictions, top=5) [0]
```

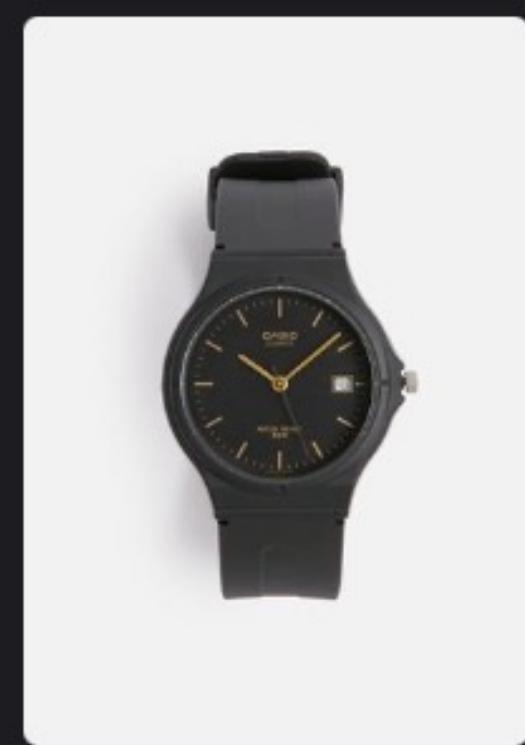


We want
to classify
into 8
specific
categories

Can't use
pretrained
model

Must train
our own

accessories	>
jackets	>
jeans	>
knitwear	>
shirts	>
shoes	>
shorts	>
tees	>
productimg_1.jpg	
productimg_6.jpg	
productimg_7.jpg	
productimg_11.jpg	
productimg_15.jpg	
productimg_26.jpg	
productimg_30.jpg	
productimg_43.jpg	
productimg_62.jpg	
productimg_81.jpg	
productimg_95.jpg	
productimg_107.jpg	
productimg_119.jpg	
productimg_126.jpg	
productimg_145.jpg	
productimg_153.jpg	
productimg_173.jpg	
productimg_176.jpg	
productimg_185.jpg	
productimg_197.jpg	
productimg_204.jpg	
productimg_219.jpg	
productimg_226.jpg	
productimg_241.jpg	



productimg_11.jpg	
JPEG image - 7 KB	
Information	Show More
Created	25 October 2017 at 23:05
Modified	25 October 2017 at 23:05
Last opened	--
Dimensions	300×432
Tags	
Add Tags...	



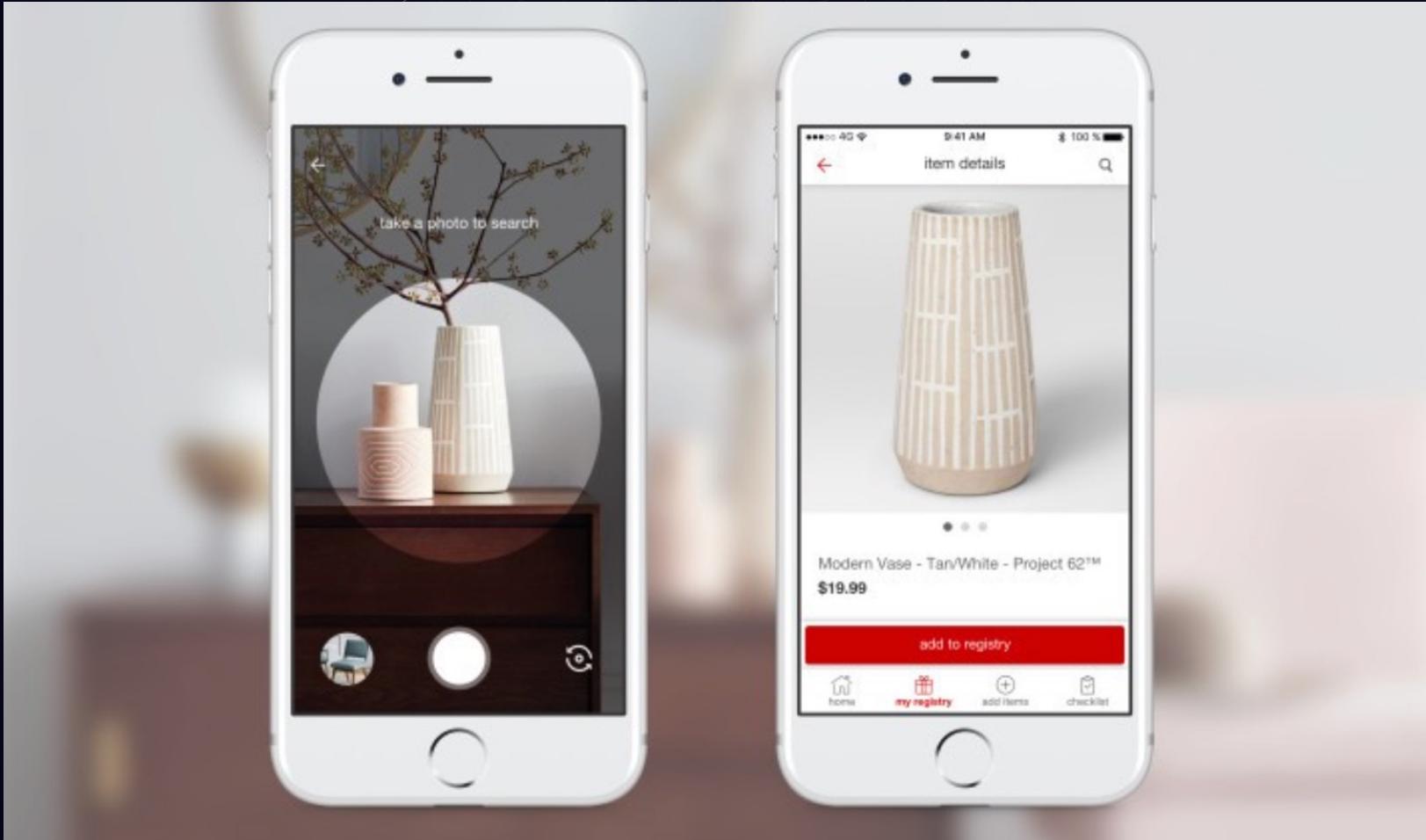
Jupyter Notebook

Let's train our own convolutional neural network
to classify product photos into one of 8 categories

Classification and Visual Search.ipynb



Visual Search



Embeddings

$f($



Sleeve
length

how
much
graphics

) = [0.14, 1.52, ...]

$f($

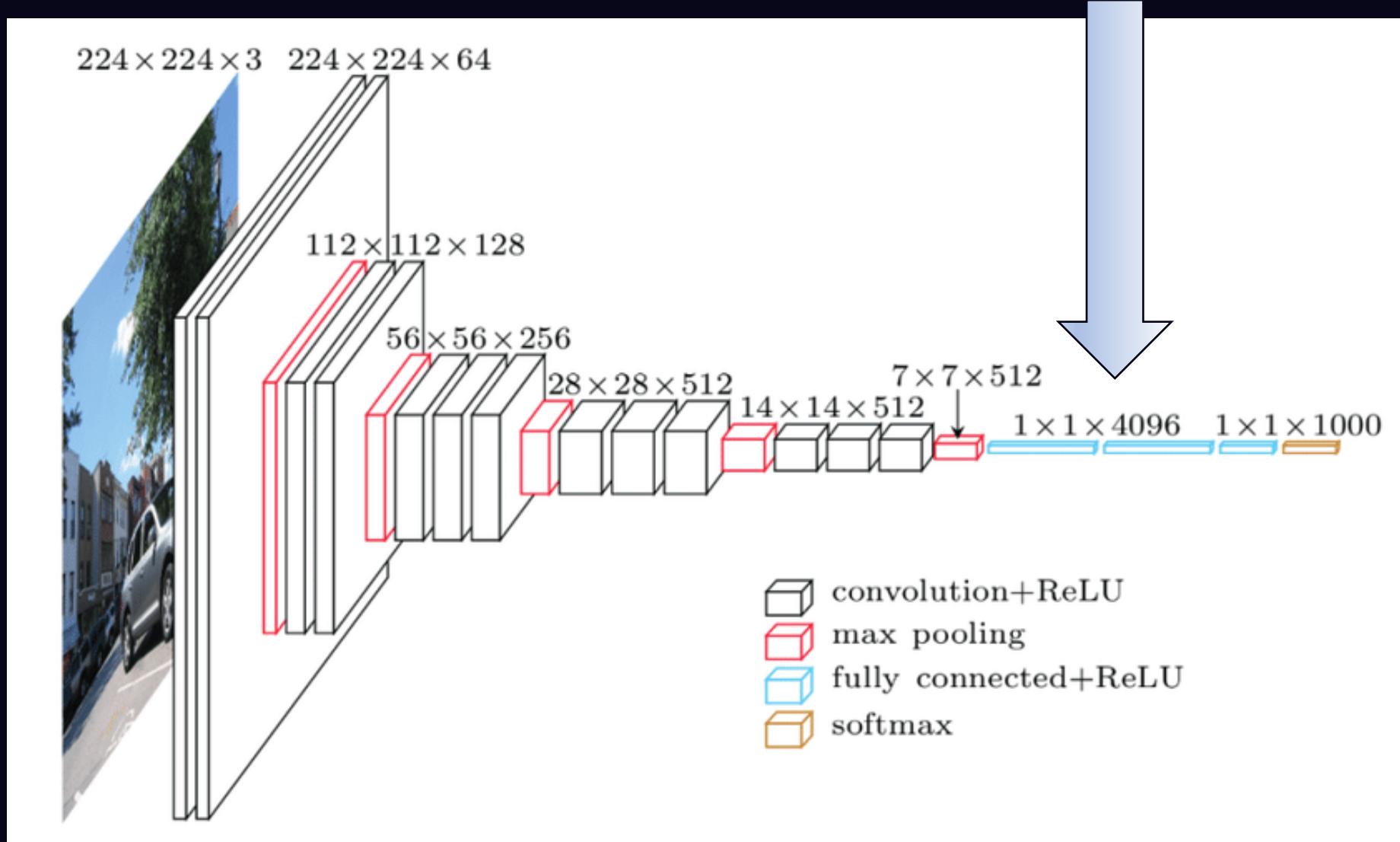


Sleeve
length

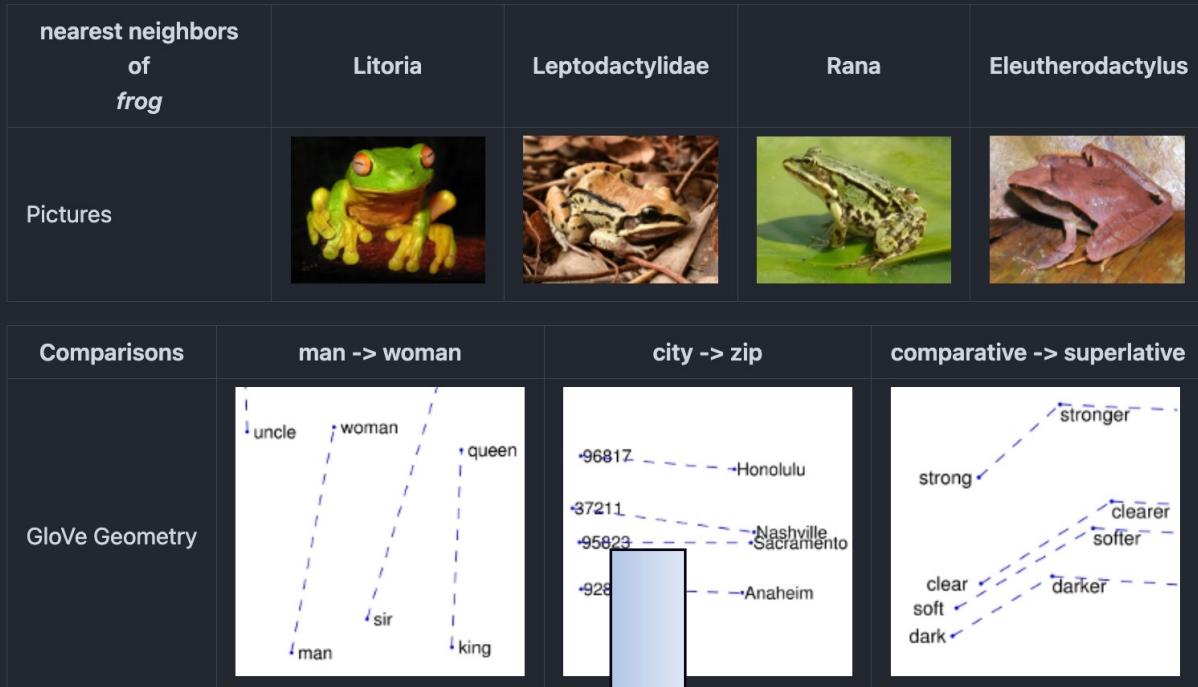
how
much
graphics

) = [2.98, 0.21, ...]

4096 dimensional image embedding



GloVe: Global Vectors for Word Representation



We provide an implementation of the GloVe model for learning word representations, and describe how to download web-dataset vectors or train your own. See the page or the paper for more information on glove vectors.

Download pre-trained word vectors

The links below contain word vectors obtained from the massive web corpora. If you want word vectors trained on massive web datasets, you need only download one of these! Pre-trained word vectors are made available under the [Public Domain Dedication and License](#).

- Common Crawl (42B tokens, 1.9M vocab, uncased, 300d vectors, 1.75 GB download): [glove.42B.300d.zip](#) [mirror]
- Common Crawl (840B tokens, 2.2M vocab, cased, 300d vectors, 2.03 GB download): [glove.840B.300d.zip](#) [mirror]
- Wikipedia 2014 + Gigaword 5 (6B tokens, 400K vocab, uncased, 300d vectors, 822 MB download): [glove.6B.zip](#) [mirror]

Packages

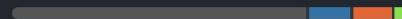
No packages published

Contributors



+ 16 contributors

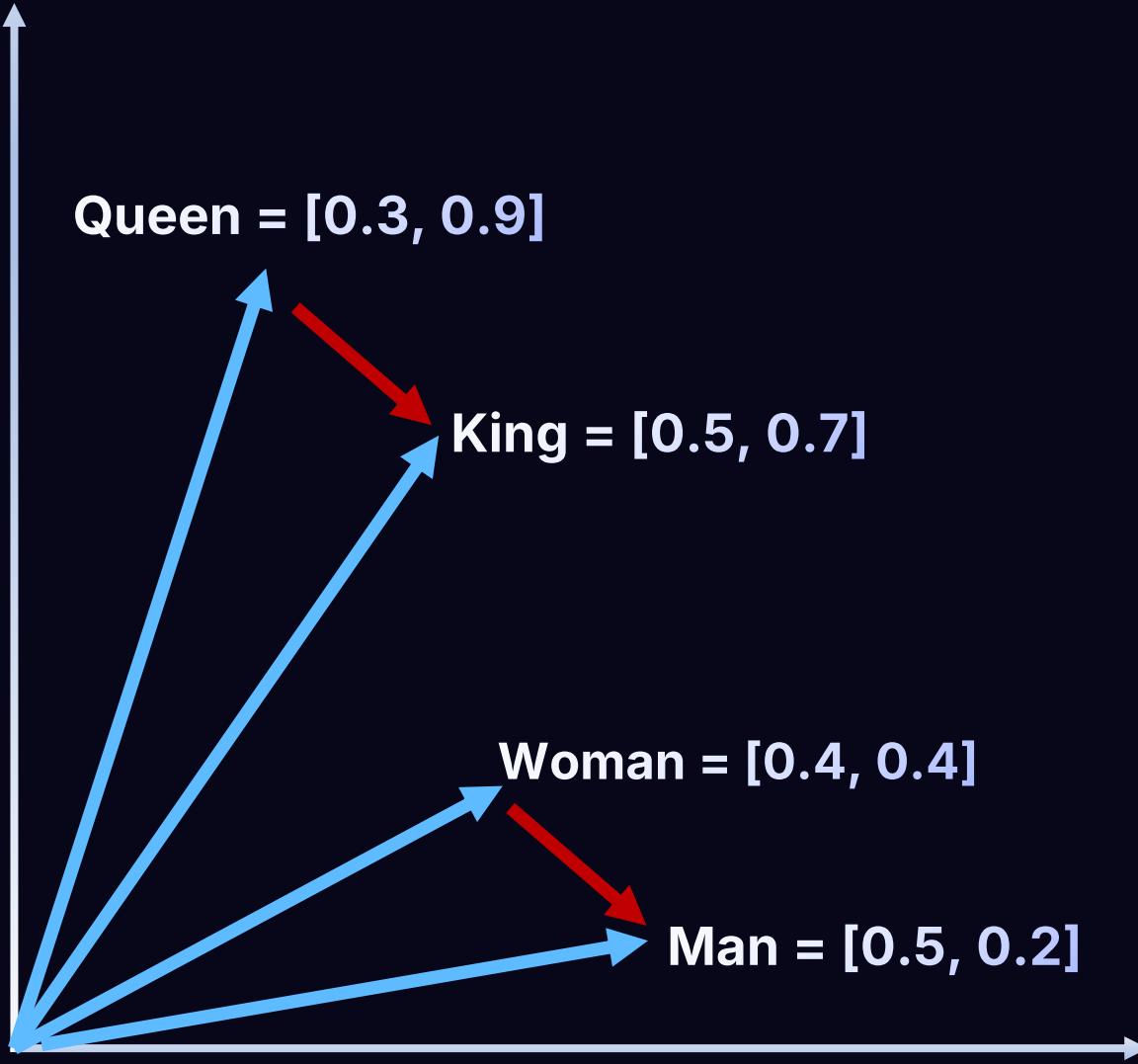
Languages



● C 70.7% ● Python 9.9%
● MATLAB 9.3% ● Shell 8.2%
● Makefile 1.9%



Word2Vec Embeddings



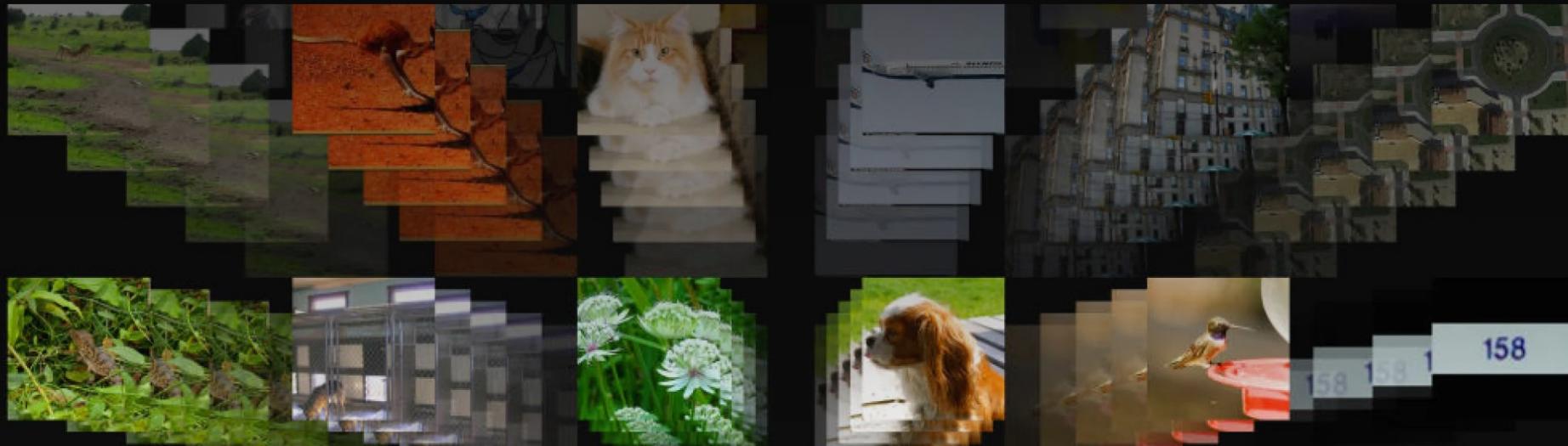
first 2 principal components of target latent factors



January 5, 2021

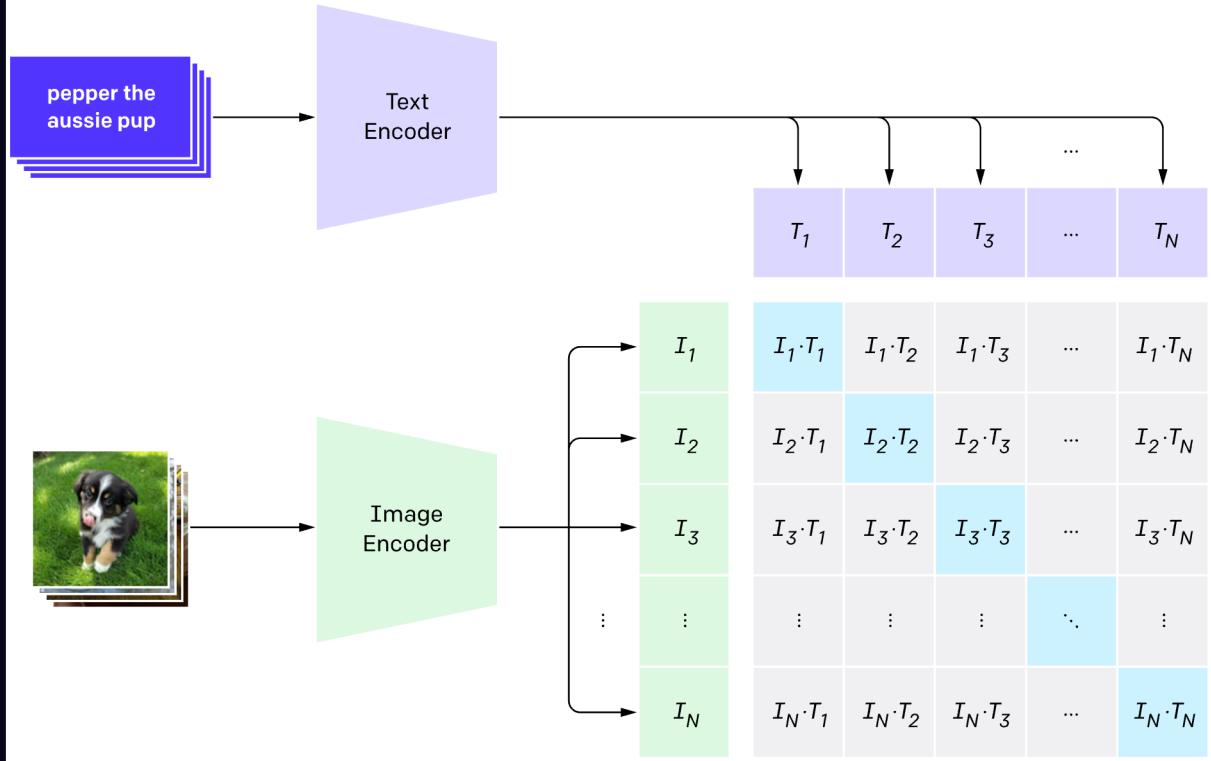
CLIP: Connecting text and images

Read paper ↗

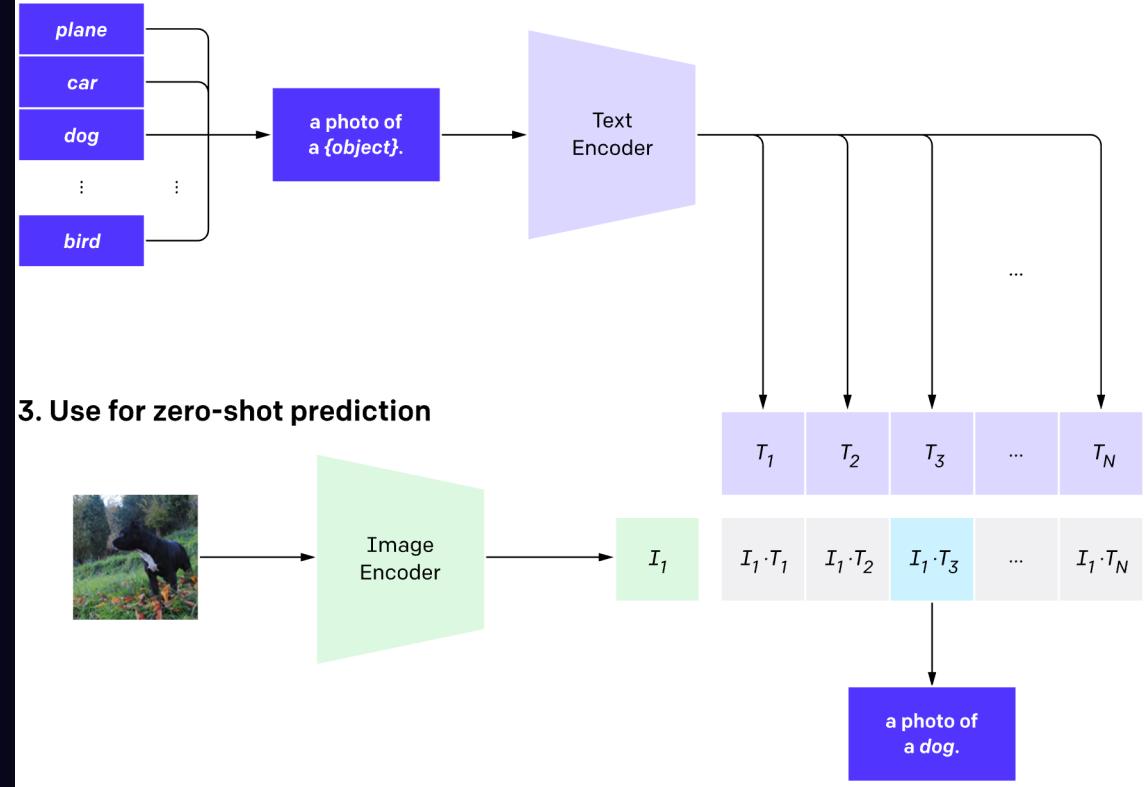


CLIP MultiModal Embeddings

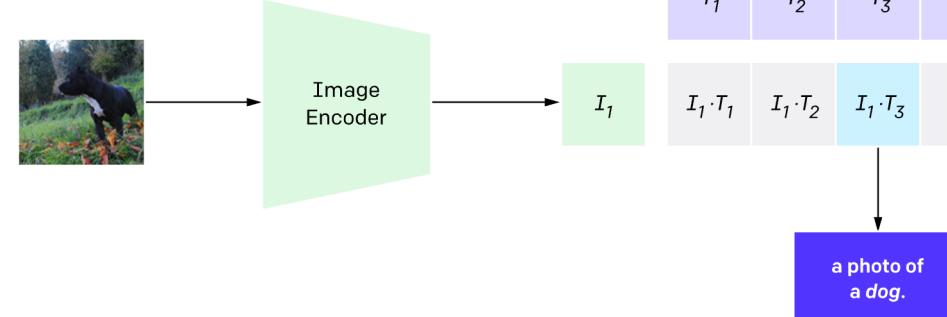
1. Contrastive pre-training



2. Create dataset classifier from label text



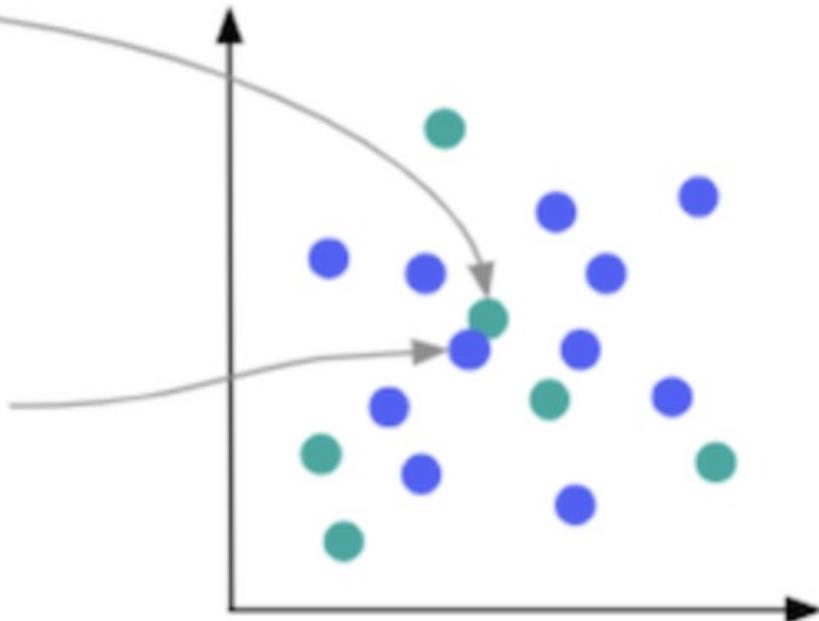
3. Use for zero-shot prediction

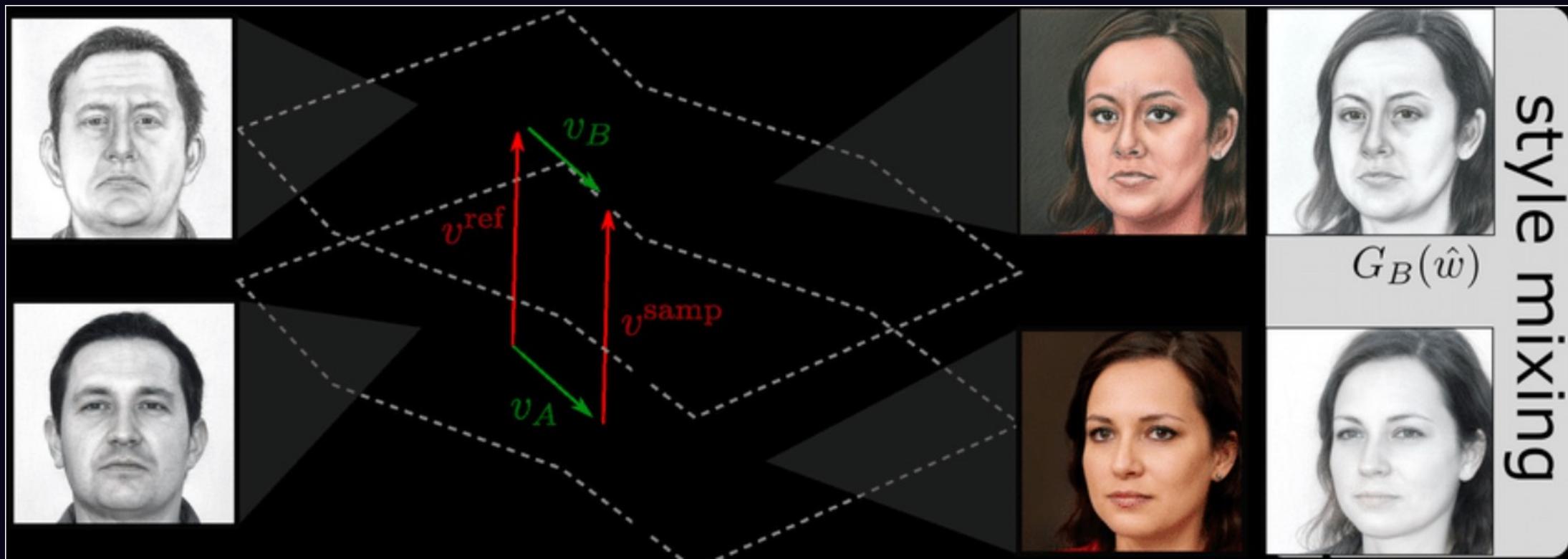
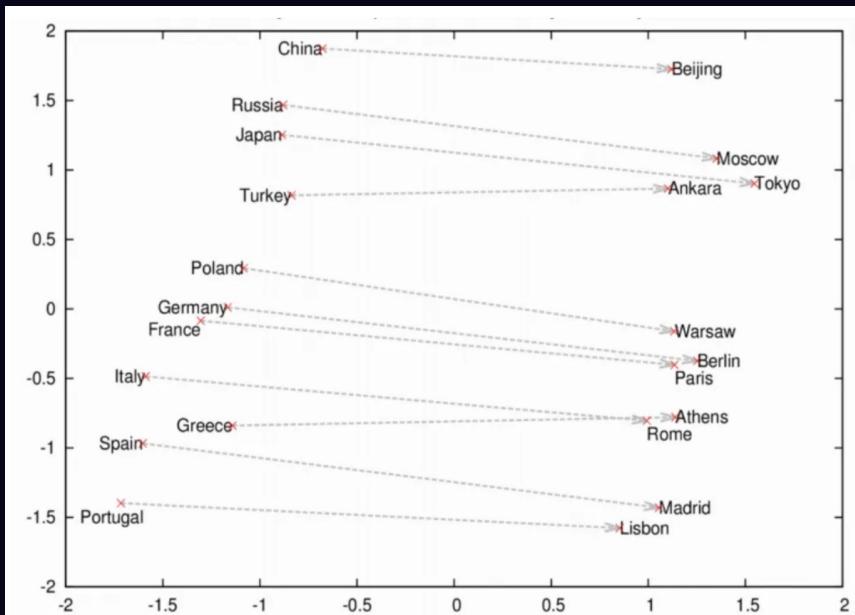




“woodblock print of the Edo period depicting three boats moving through a storm-tossed sea with a large wave forming a spiral in the centre and Mount Fuji visible in the background”

Joint embedding space
(typically a vector space of dimension 512 or 768)





Jupyter Notebook

Let's do some visual search!

Classification and Visual Search.ipynb



Input Image



Top 10 Similar Images

Rank 1



Rank 2



Rank 3



Rank 4



Rank 5



Rank 6



Rank 7



Rank 8



Rank 9



Rank 10



Jupyter Notebook

Let's detect objects within images
Object Detection.ipynb



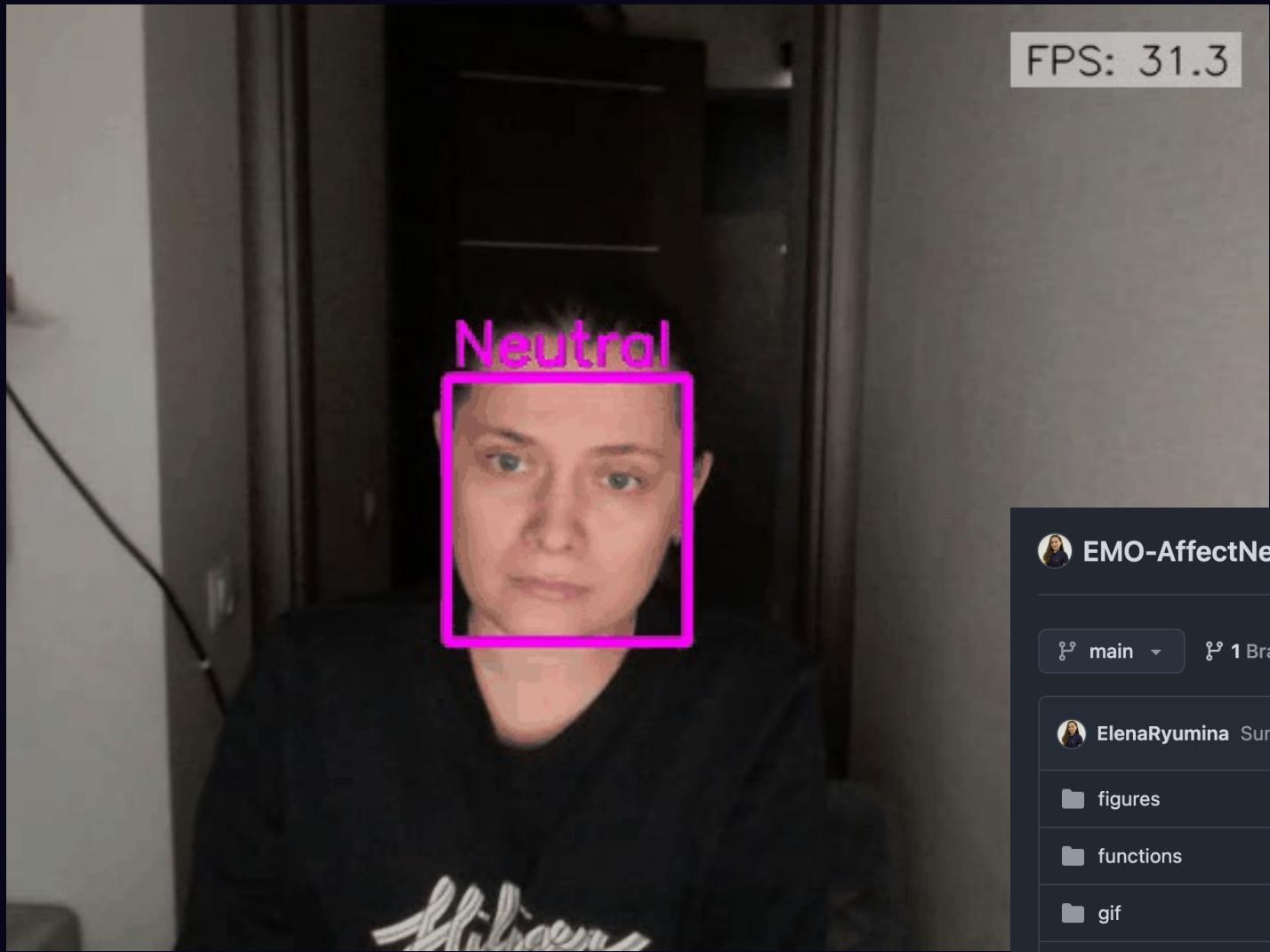
Original Image



Detected 14 persons
Detected 4 cars

YOLOv8 Object Detection





 **EMO-AffectNetModel** Public

 main   1 Branch  0 Tags  Go to file 

 ElenaRyumina	Summary	3dc8ed2 ·
 figures	Update results_emo_pred_videos.PNG	
 functions	Summary	
 gif	Update	
 LICENSE	Initial commit	
 README.md	Update README.md	
 check_backbone_models_by_webcam.ipynb	Update	

<https://github.com/ElenaRyumina/EMO-AffectNetModel>



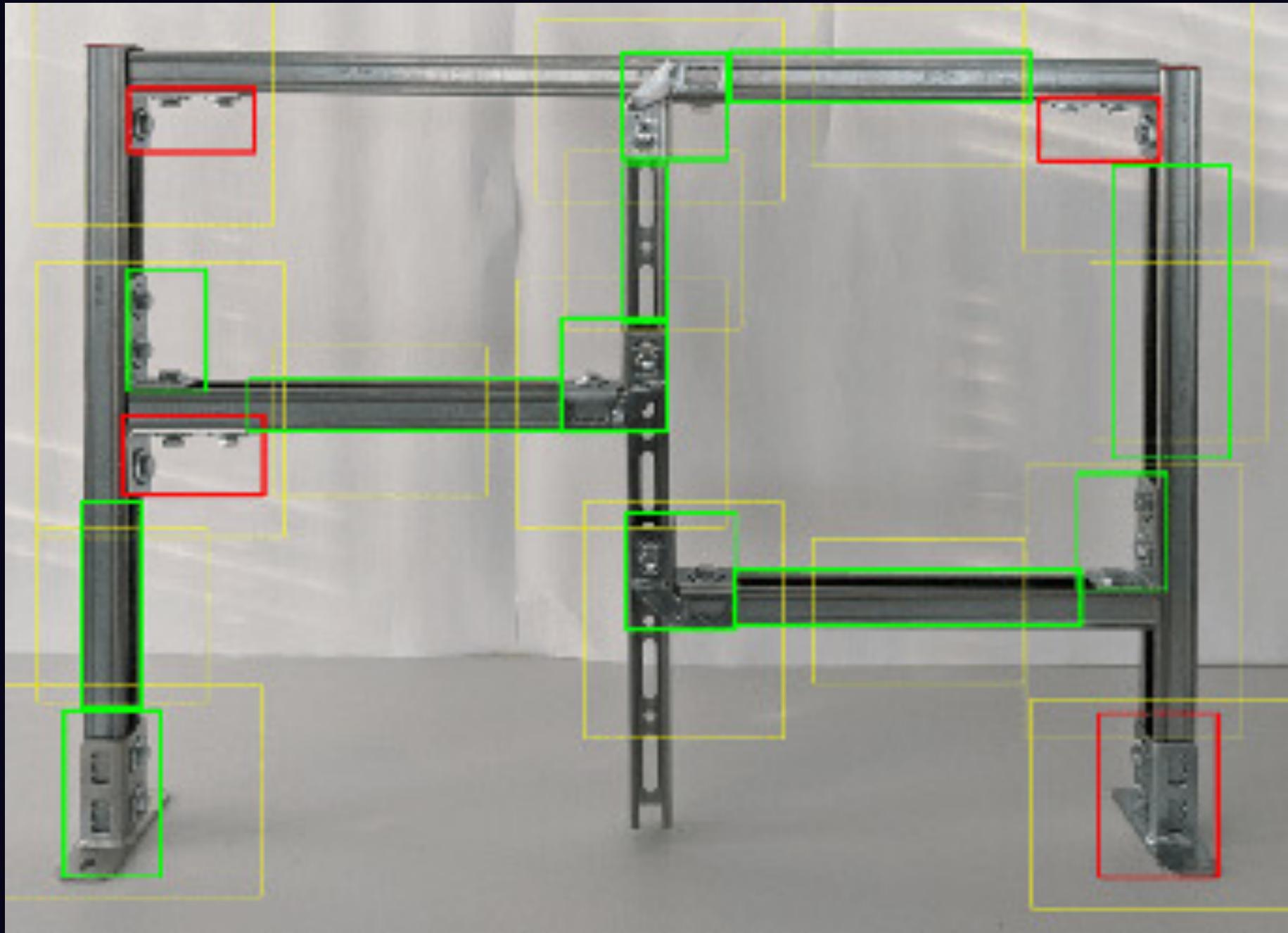
13:56:12

1



O





Jupyter Notebook

Let's segment objects within images
Segmentation.ipynb



Original Image



Segmentation Overlay



Original Image



Original Image with 50% Transparent Segmentation Mask



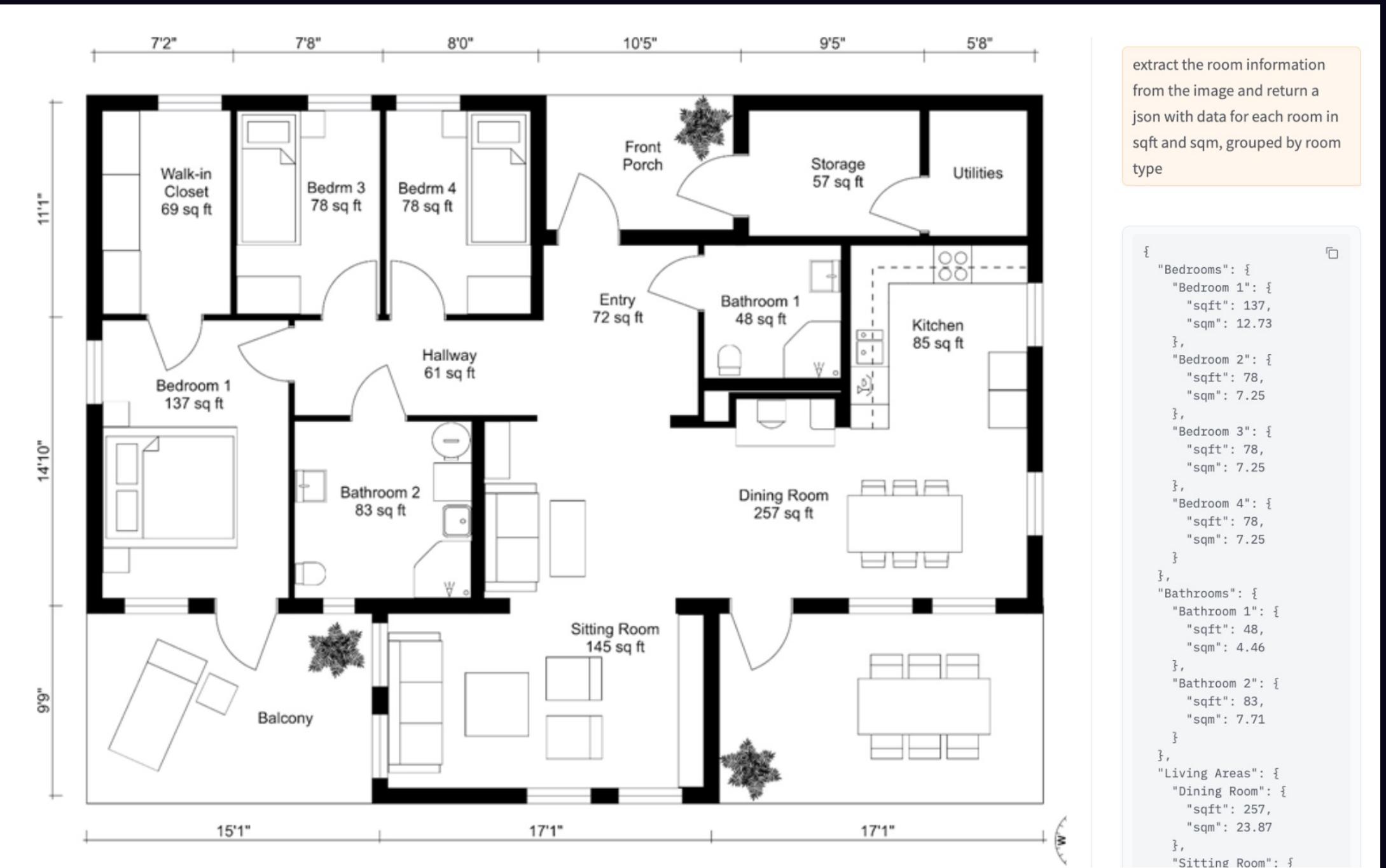
Car with Drop Shadow on New Background



MultiModal LLMs Are Cool

<https://huggingface.co/spaces/Qwen/Qwen2-VL>





extract the room information from the image and return a json with data for each room in sqft and sqm, grouped by room type

```
{
  "Bedrooms": {
    "Bedroom 1": {
      "sqft": 137,
      "sqm": 12.73
    },
    "Bedroom 2": {
      "sqft": 78,
      "sqm": 7.25
    },
    "Bedroom 3": {
      "sqft": 78,
      "sqm": 7.25
    },
    "Bedroom 4": {
      "sqft": 78,
      "sqm": 7.25
    }
  },
  "Bathrooms": {
    "Bathroom 1": {
      "sqft": 48,
      "sqm": 4.46
    },
    "Bathroom 2": {
      "sqft": 83,
      "sqm": 7.71
    }
  },
  "Living Areas": {
    "Dining Room": {
      "sqft": 257,
      "sqm": 23.87
    },
    "Sitting Room": {
      "sqft": 145,
      "sqm": 13.41
    }
  }
}
```





Directions: For adults, take two (2) vegan capsules daily, preferably with a meal. Do not exceed stated dose.

Nutrition Information

Typically Per Daily Dose

Psyllium Husk Powder 1400mg

Ingredients: Psyllium Husk Powder, Capsule Shell (Hydroxypropylmethylcellulose), Anti-Caking Agents (Magnesium Salts of Fatty Acids, Silicon Dioxide).

May contain Sesame Seeds & Mustard. For allergens, see the ingredients in bold.

Notice: Take this product with 220ml of fluids. Taking this product without adequate fluid may cause the possibility of choking. Do not use this product if you have difficulty swallowing. If you experience chest pain, vomiting or difficulty in swallowing or breathing after taking this product, seek immediate medical attention. Do not take before sleeping. Fibre products can affect the absorption of many medications. Do not take this product within 1.5 hours of taking medication.

Free From: Artificial Colour, Artificial Flavour, Artificial Sweetener, Preservatives.

SUITABLE FOR VEGETARIANS & VEGANS

HU16790 C

Horbäach®

return a JSON object with all the data about the product in this image including its name and manufacturer

```
{  
  "name": "Psyllium Husk",  
  "manufacturer": "Horbaach",  
  "strength": "1400mg",  
  "type": "Plantago Ovata Plant Seeds",  
  "quantity": "365 Vegan Capsules",  
  "directions": "For adults, take two (2) vegan capsules daily, preferably with a meal.",  
  "nutrition_information": {  
    "per_daily_dose": {  
      "psyllium_husk_powder": "1400mg"  
    },  
    "ingredients": [  
      "Psyllium Husk Powder",  
      "Capsule Shell (Hydroxypropylmethylcellulose)",  
      "Anti-Caking Agents (Magnesium Salts of Fatty Acids, Silicon Dioxide)"  
    ],  
    "allergen_information": "May contain Sesame Seeds & Mustard. For allergens, see the ingredients in bold.",  
    "notice": "Take this product with 220ml of fluids. Taking this product without adequate fluid may cause the possibility of choking.",  
    "free_from": [  
      "Artificial Colour",  
      "Artificial Flavour",  
      "Artificial Sweetener",  
      "Preservatives"  
    ],  
    "suitable_for": [  
      "Vegetarians",  
      "Vegans"  
    ]  
  }  
}
```



Qwen2-VL-Max

This WebUI is based on Qwen2-VL-Max, developed by Alibaba Cloud.

本WebUI基于Qwen2-VL-Max。

Qwen2-VL-Max

return a JSON object with all the data from the passport

```
{
  "passport": {
    "type": "P",
    "code": "VNM",
    "number": "B9575757",
    "holder": {
      "full_name": "PHAM KIEU M SURAPAPAN",
      "nationality": "VIET NAM / VIETNAMESE",
      "date_of_birth": "22/02/1976",
      "sex": "F",
      "place_of_birth": "TAM BI"
    },
    "issue": {
      "date": "28/12/2008",
      "place": "Cục Quản lý xuất nhập cảnh"
    },
    "expiry": "28/12/2018",
    "id_card_number": "B9575757"
  }
}
```



Here's an example of my normal handwriting. Notice how letters are connected in a way that has little to do with word length - as in "normal" - 1, 2, 3 letters are grouped.

write out all the text in this handwritten note. be careful not to make any mistakes.

Here's an example of my normal handwriting. Notice how letters are connected in a way that has little to do with word length - as in "normal" - 1, 2, 3 letters are grouped.





Hugging Face

Computer Vision



Depth Estimation



Image Classification



Object Detection



Image Segmentation



Text-to-Image



Image-to-Text



Image-to-Image



Image-to-Video



Unconditional Image Generation



Video Classification



Text-to-Video



Zero-Shot Image Classification



Mask Generation



Zero-Shot Object Detection



Text-to-3D



Image-to-3D



Image Feature Extraction



Keypoint Detection



Speech Data





How to learn Python programming | Guido van Rossum and Lex Fridman

Lex
Clips

Lex Clips 1.35M subscribers

Subscribe

14K



Share

Download

...

504K views 1 year ago

Lex Fridman Podcast full episode: Guido van Rossum: Python and the Future



```
~/Documents/_meetups/pyconza 2024/code/Speech/data git:(main)±4 (1.982s)
ffmpeg -i lex_guido.mp3 -f segment -segment_time 30 -c copy -reset_timestamps 1 output%03d.mp3
```

```
ffmpeg version 7.0.2 Copyright (c) 2000-2024 the FFmpeg developers
  built with Apple clang version 15.0.0 (clang-1500.3.9.4)
configuration: --prefix=/opt/homebrew/Cellar/ffmpeg/7.0.2_1 --enable-shared --enable-pthreads --enable-version3 --cc=clang --host-cflags= --host-ldflags=' -Wl,-ld_classic' --enable-ffplay --enable-gnutls --enable-libaom --enable-libarbb24 --enable-libbluray --enable-libdav1d --enable-libharfbuzz --enable-libjxl --enable-libmp3lame --enable-libopus --enable-librav1e --enable-librist --enable-librubberband --enable-libsnappy --enable-libsrt --enable-libssh --enable-libsrtav1 --enable-libtesseract --enable-libtheora --enable-libvidstab --enable-libvmaf --enable-libvorbis --enable-libvpx --enable-libwebp --enable-libx264 --enable-libx265 --enable-libxml2 --enable-libxvid --enable-lzma --enable-libfontconfig --enable-libfreetype --enable-frei0r --enable-libass --enable-libopencore-amrnb --enable-libopencore-amrwb --enable-libopenjpeg --enable-libspeex --enable-libsoxr --enable-libzmq --enable-libzimg --disable-libjack --disable-indev=jack --enable-videotoolbox --enable-audiotoolbox --enable-neon
libavutil      59. 8.100 / 59. 8.100
libavcodec     61. 3.100 / 61. 3.100
libavformat    61. 1.100 / 61. 1.100
libavdevice    61. 1.100 / 61. 1.100
libavfilter     10. 1.100 / 10. 1.100
libswscale      8. 1.100 /  8. 1.100
libswresample   5. 1.100 /  5. 1.100
libpostproc    58. 1.100 / 58. 1.100
Input #0, mp3, from 'lex_guido.mp3':
Metadata:
  major_brand     : isom
  minor_version   : 512
  compatible_brands: isomiso2avc1mp41
  encoder         : Lavf60.3.100
Duration: 00:07:06.16, start: 0.025057, bitrate: 128 kb/s
Stream #0:0: Audio: mp3 (mp3float), 44100 Hz, stereo, fltp, 128 kb/s
Metadata:
  encoder         : Lavc60.3.
Stream mapping:
  Stream #0:0 -> #0:0 (copy)
[segment @ 0x156004080] Opening 'output000.mp3' for writing
Output #0, segment, to 'output%03d.mp3':
Metadata:
  major_brand     : isom
  minor_version   : 512
  compatible_brands: isomiso2avc1mp41
  encoder         : Lavf61.1.100
Stream #0:0: Audio: mp3, 44100 Hz, stereo, fltp, 128 kb/s
Metadata:
  encoder         : Lavc60.3.
Press [q] to stop, [?] for help
[segment @ 0x156004080] Opening 'output001.mp3' for writing
[segment @ 0x156004080] Opening 'output002.mp3' for writing
[segment @ 0x156004080] Opening 'output003.mp3' for writing
```



mp3s



- output000.mp3
- output001.mp3
- output002.mp3
- output003.mp3
- output004.mp3
- output005.mp3
- output006.mp3
- output007.mp3
- output008.mp3
- output009.mp3
- output010.mp3
- output011.mp3
- output012.mp3
- output013.mp3
- output014.mp3



Jupyter Notebook

Transcribe.ipynb



large model

```
[32]: transcribe_audio_file("data/mp3s/output000.mp3", model)
```



```
Start time: 2024-10-02 14:43:13.105156
```

```
/Users/alexc/Documents/_meetups/pyconza 2024/code/venv/lib/python3.11/site-packages/whisper/transcribe.py:126: UserWarning: FP16 is not supported on CPU; using FP32 instead
    warnings.warn("FP16 is not supported on CPU; using FP32 instead")
```

```
Stop time: 2024-10-02 14:43:30.313052
```

```
17.21 seconds taken
```

```
Transcribed data/mp3s/output000.mp3
```

```
Do you have advice for a programming beginner on how to learn Python the right way? Find something you actually want to do with it. If you say, I want to learn skill X, that's not enough motivation. You need to pick something and it can be...
```

```
{
    "text": " Do you have advice for a programming beginner on how to learn Python the right way? Find something you actually want to do with it. If you say, I want to learn skill X, that's not enough motivation. You need to pick something and it can be...",  

    "segments": [
```

```
17.21 seconds taken - still fast enough to do "in real time"
```



small model

```
[33]: transcribe_audio_file("data/mp3s/output000.mp3", model_small)
```



Start time: 2024-10-02 14:43:30.317192

```
/Users/alexc/Documents/_meetups/pyconza 2024/code/venv/lib/python3.11/site-packages/whisper/transcribe.py:126: UserWarning: FP16 is not supported on CPU; using FP32 instead  
    warnings.warn("FP16 is not supported on CPU; using FP32 instead")
```

Stop time: 2024-10-02 14:43:33.806880

3.49 seconds taken

Transcribed data/mp3s/output000.mp3

Do you have advice for a programming beginner on how to learn Python the right way? Find something you actually want to do with it. If you say, I want to learn skill X, that's not enough motivation. You need to pick something and it can be a...

```
{  
    "text": " Do you have advice for a programming beginner on how to learn Python the right way? Find something you actually want to do with it. If you say, I want to learn skill X, that's not enough motivation. You need to pick something and it can be a...",  
    "segments": [  
        {"text": " Do you have advice for a programming beginner on how to learn Python the right way? Find something you actually want to do with it. If you say, I want to learn skill X, that's not enough motivation. You need to pick something and it can be a...", "start": 0, "end": 1000},  
        {"text": "...", "start": 1000, "end": 1000}  
    ]  
}
```

3.49 seconds taken

Note 1: the same transcription results but 5 times slower with large model

Note 2: small transcription model is transcribing around 10 seconds of audio per second



Homework

Jupyter Notebook

Speech emotion recognition.ipynb

<https://github.com/mbamvianney/SPEECH-EMOTION-RECOGNITION-WITH-MACHINE-LEARNING>





Hugging Face

Audio



Text-to-Speech



Text-to-Audio



Automatic Speech Recognition



Audio-to-Audio



Audio Classification

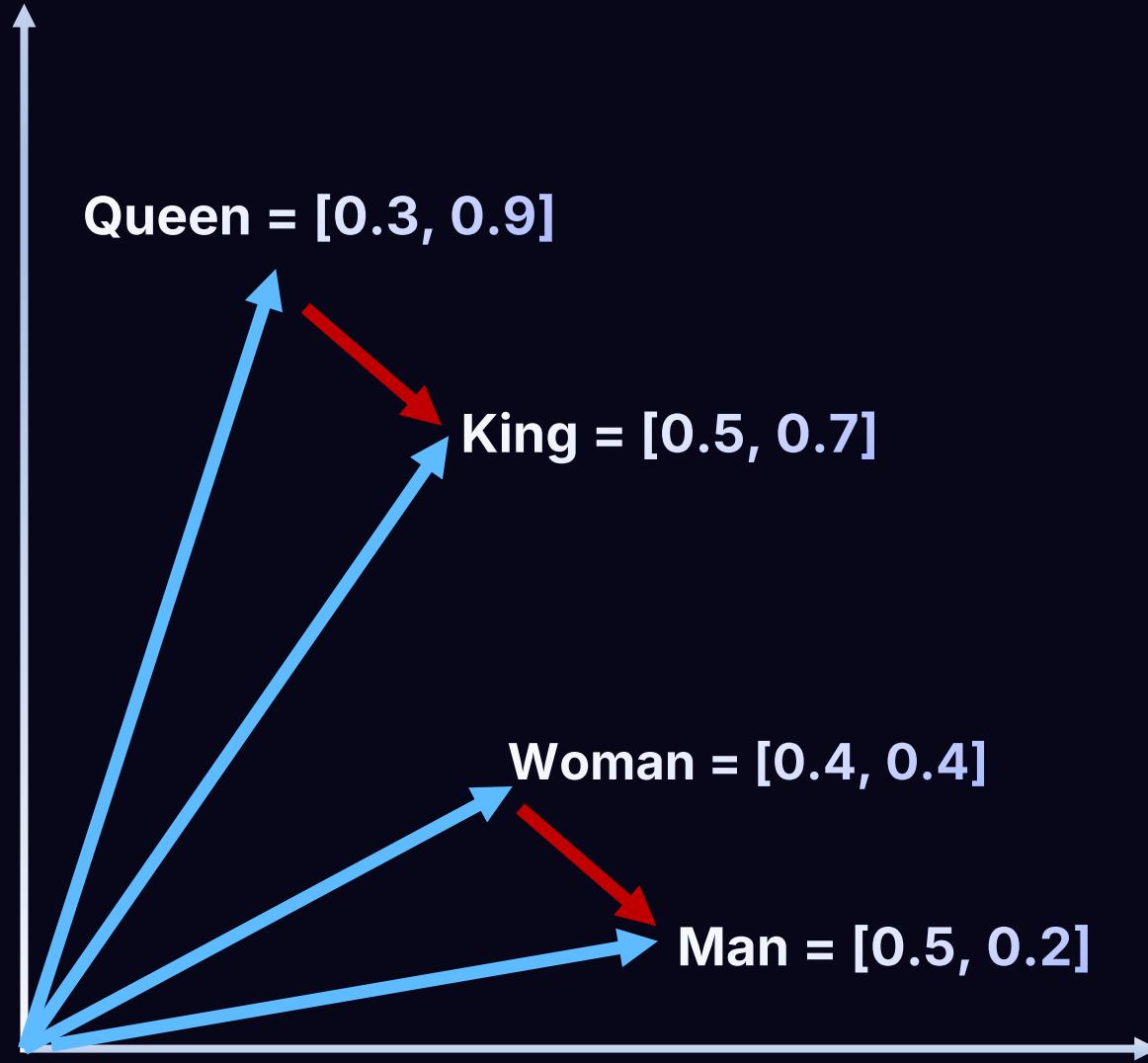


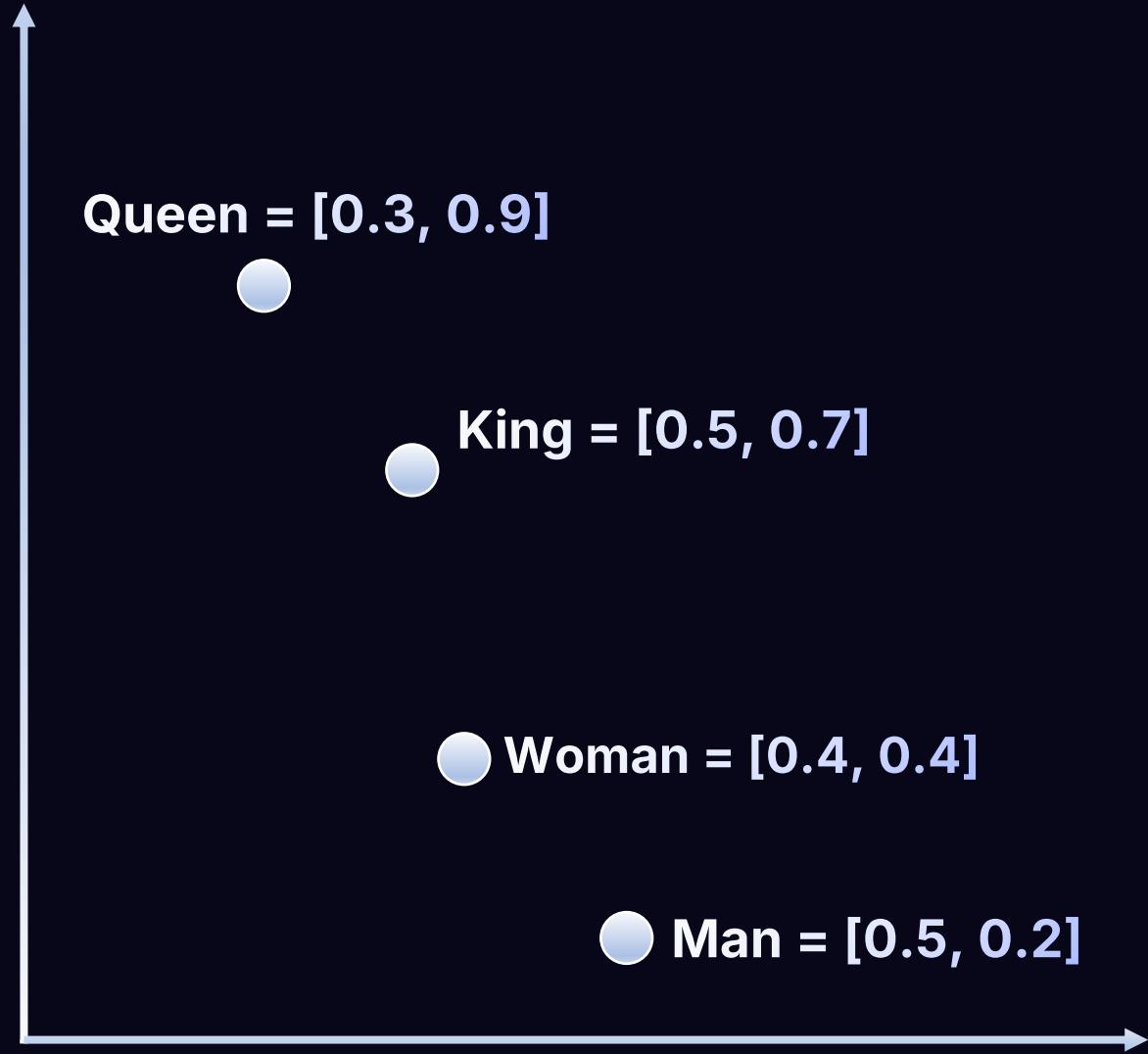
Voice Activity Detection

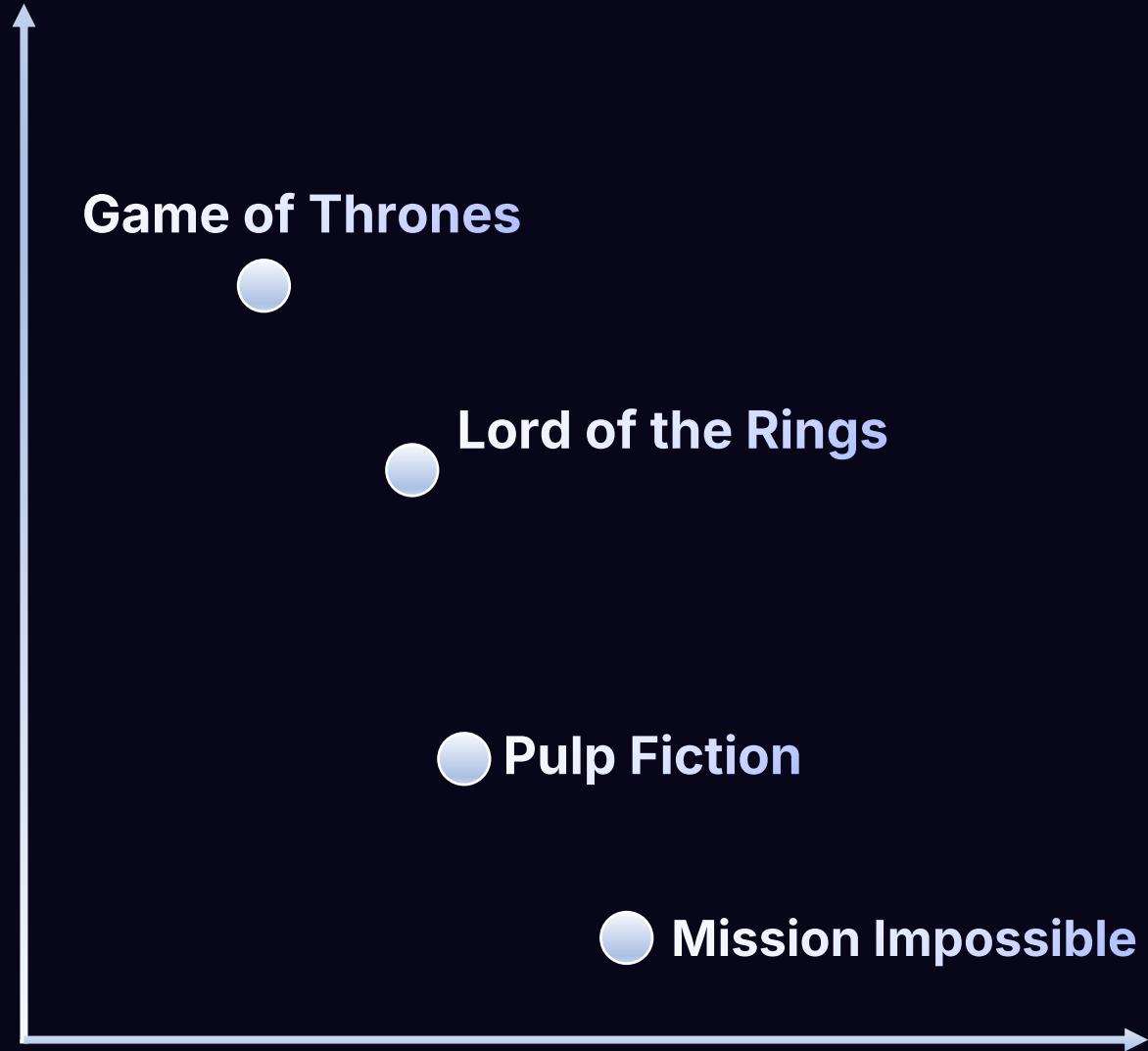


Text Data









Embeddings

$f($



Sleeve
length

how
much
graphics

) = [0.14, 1.52, ...]

$f($



Sleeve
length

how
much
graphics

) = [2.98, 0.21, ...]

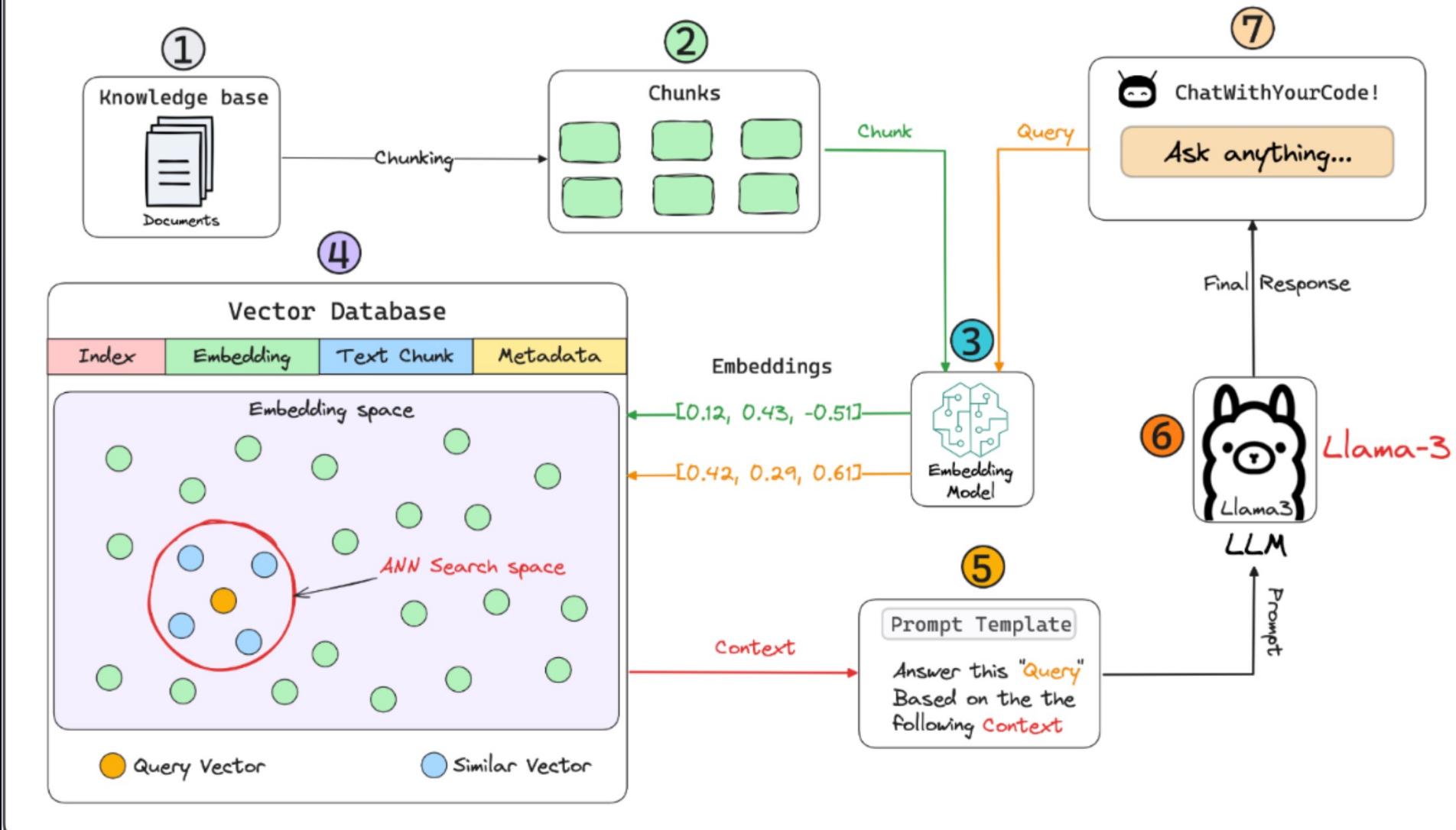
Embeddings

$f(\boxed{\text{Python is
the best
language for
AI}}) = [2.41, 1.79, \dots]$

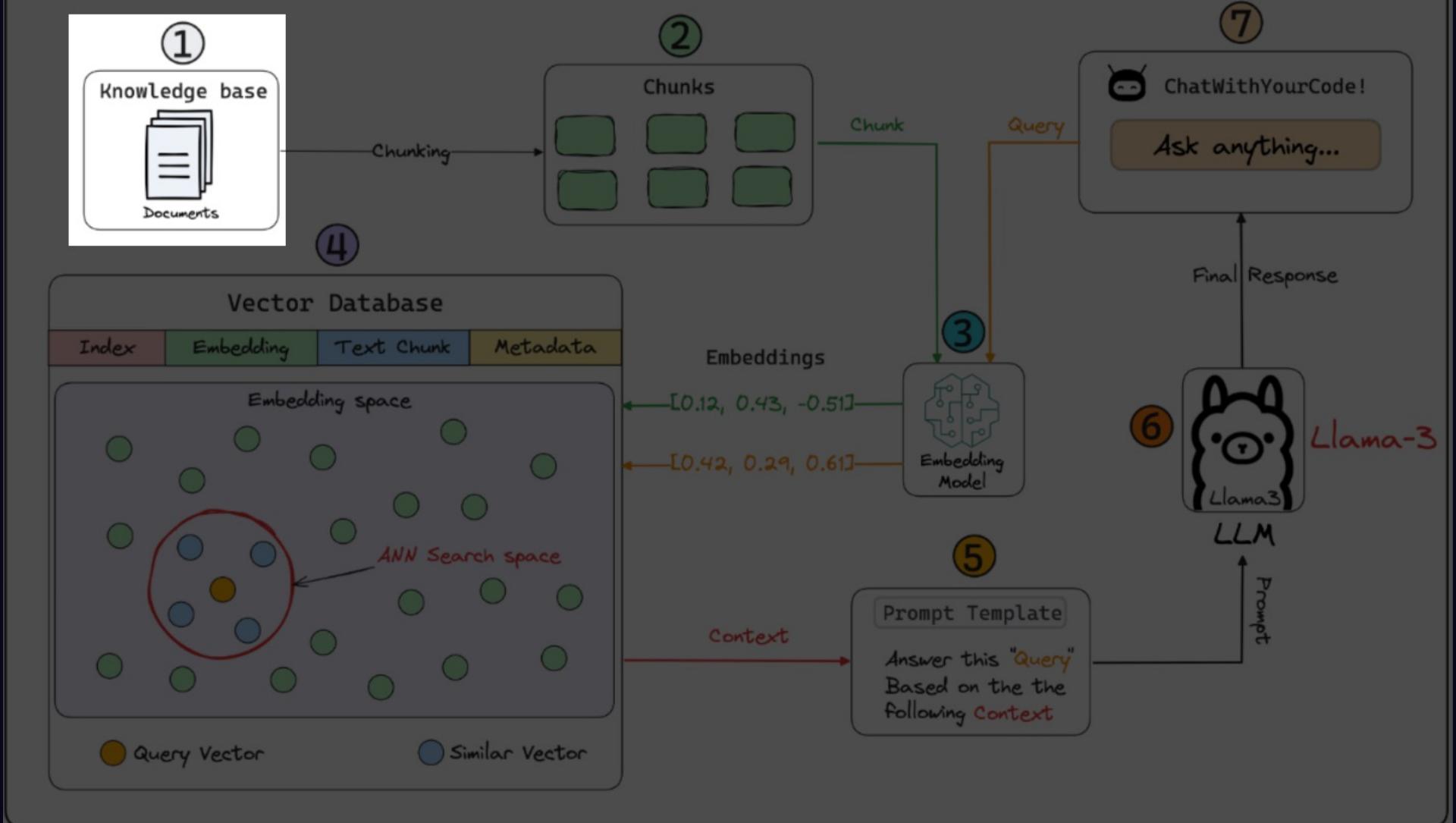
Tech topic Casual tone



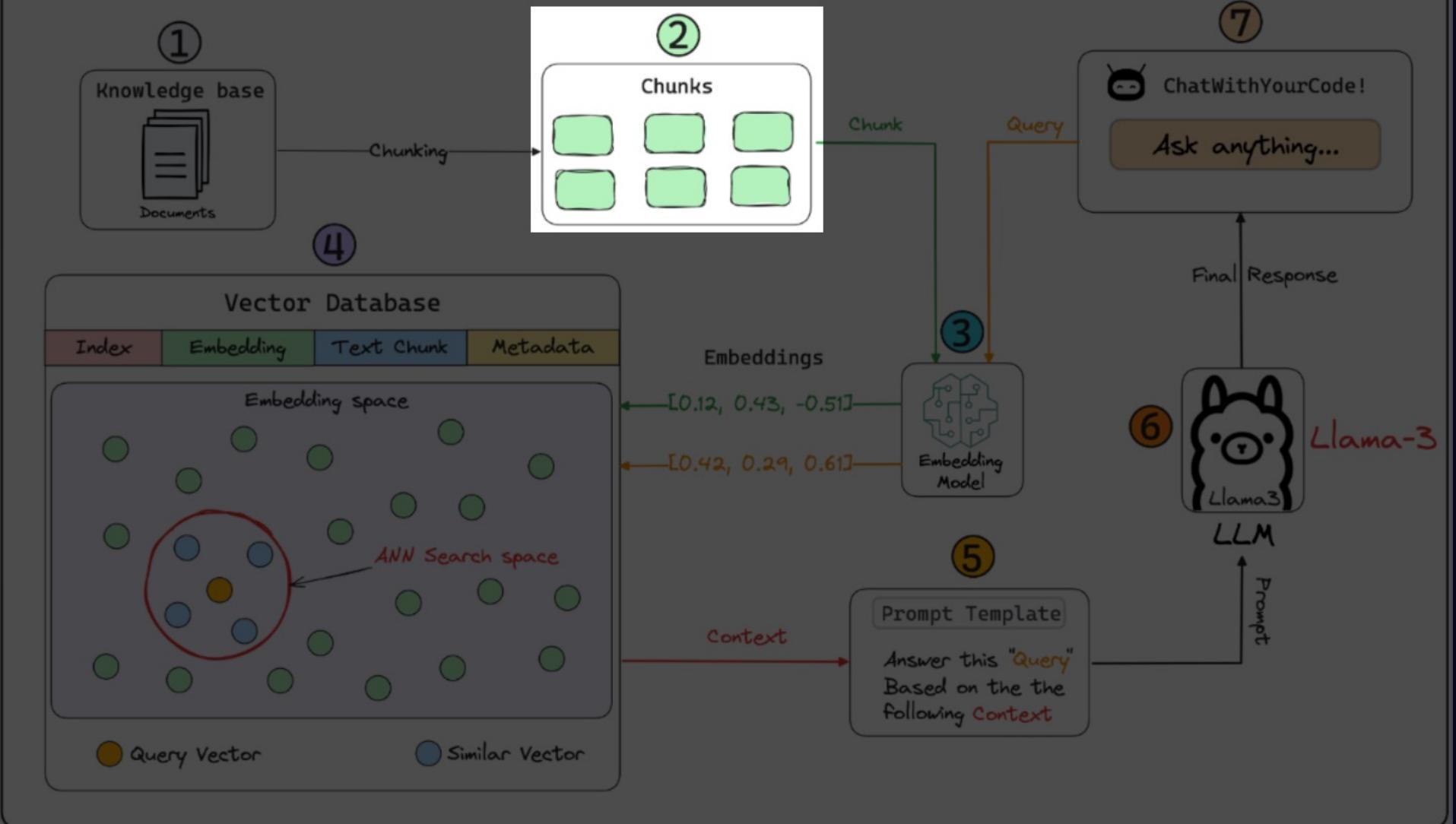
Document Chat RAG using Llama-3!



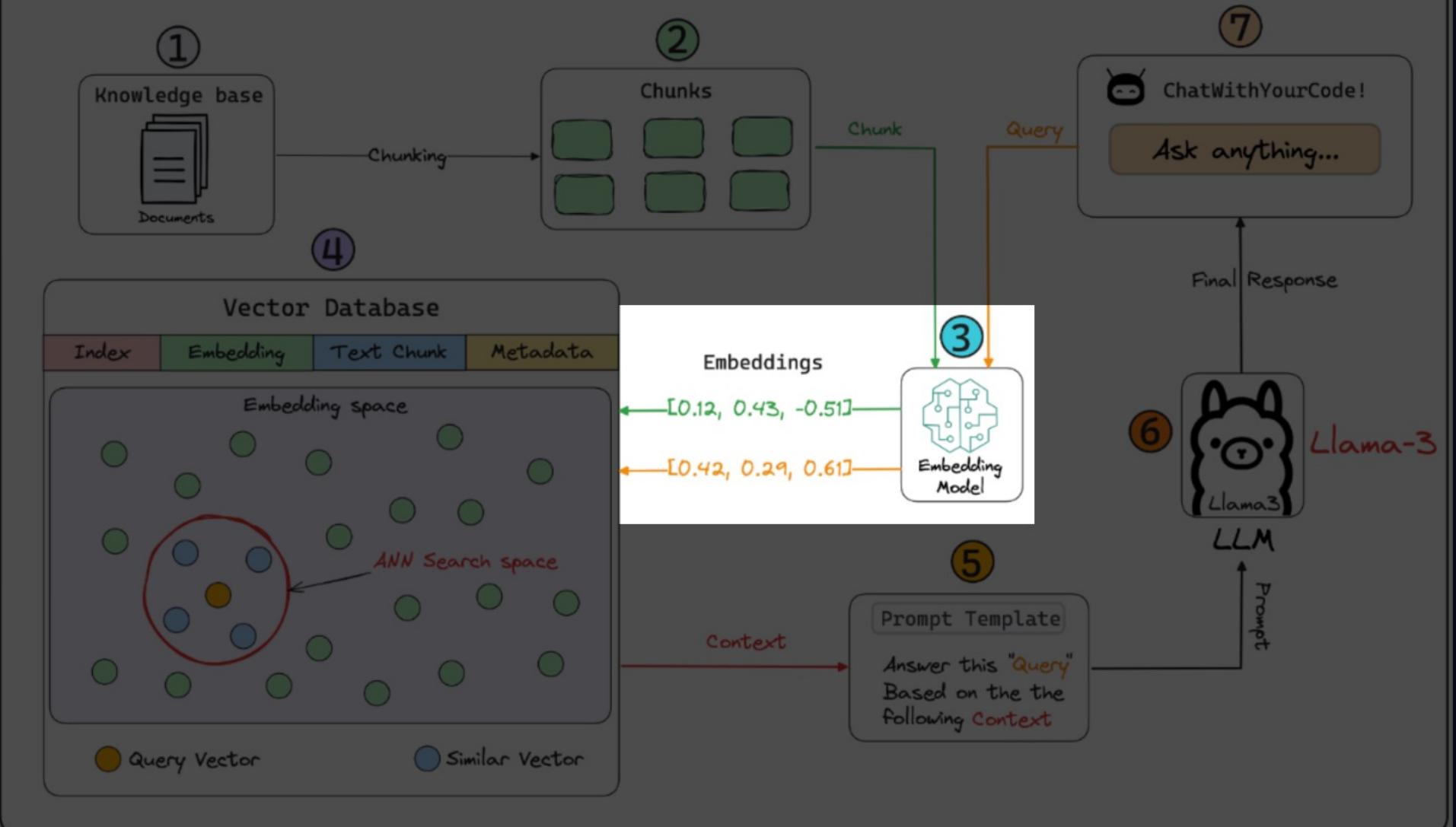
Document Chat RAG using Llama-3!



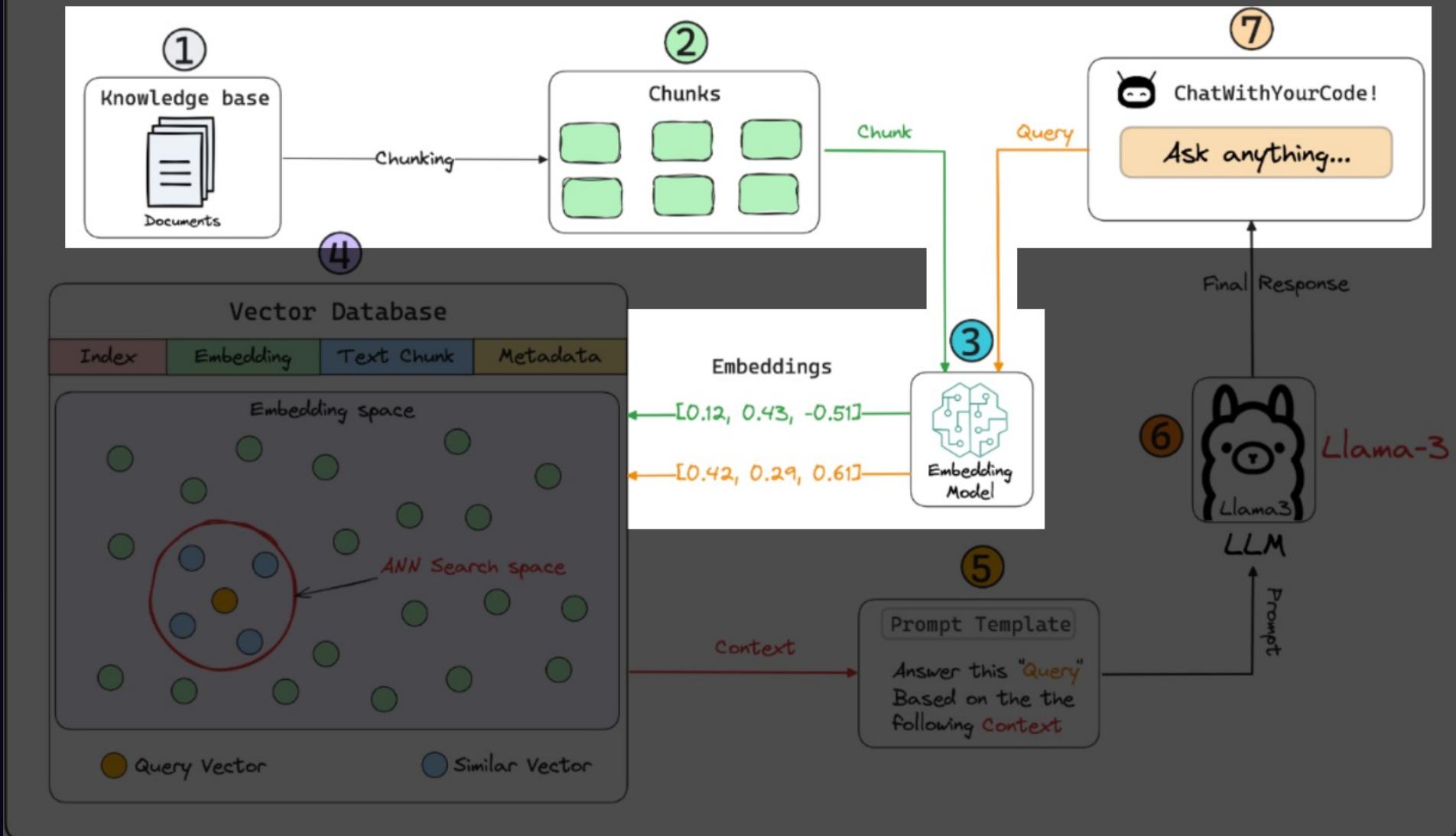
Document Chat RAG using Llama-3!



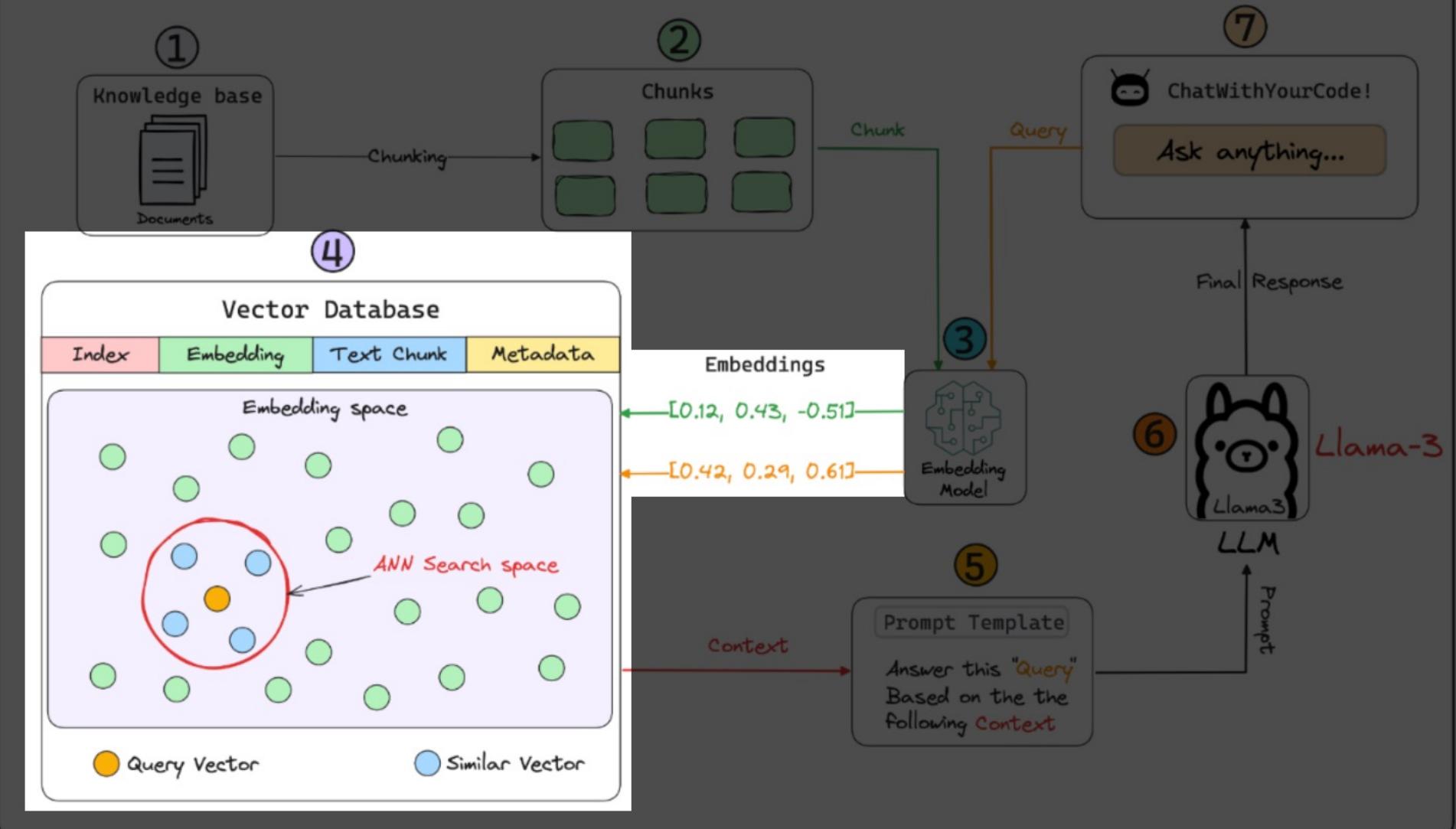
Document Chat RAG using Llama-3!



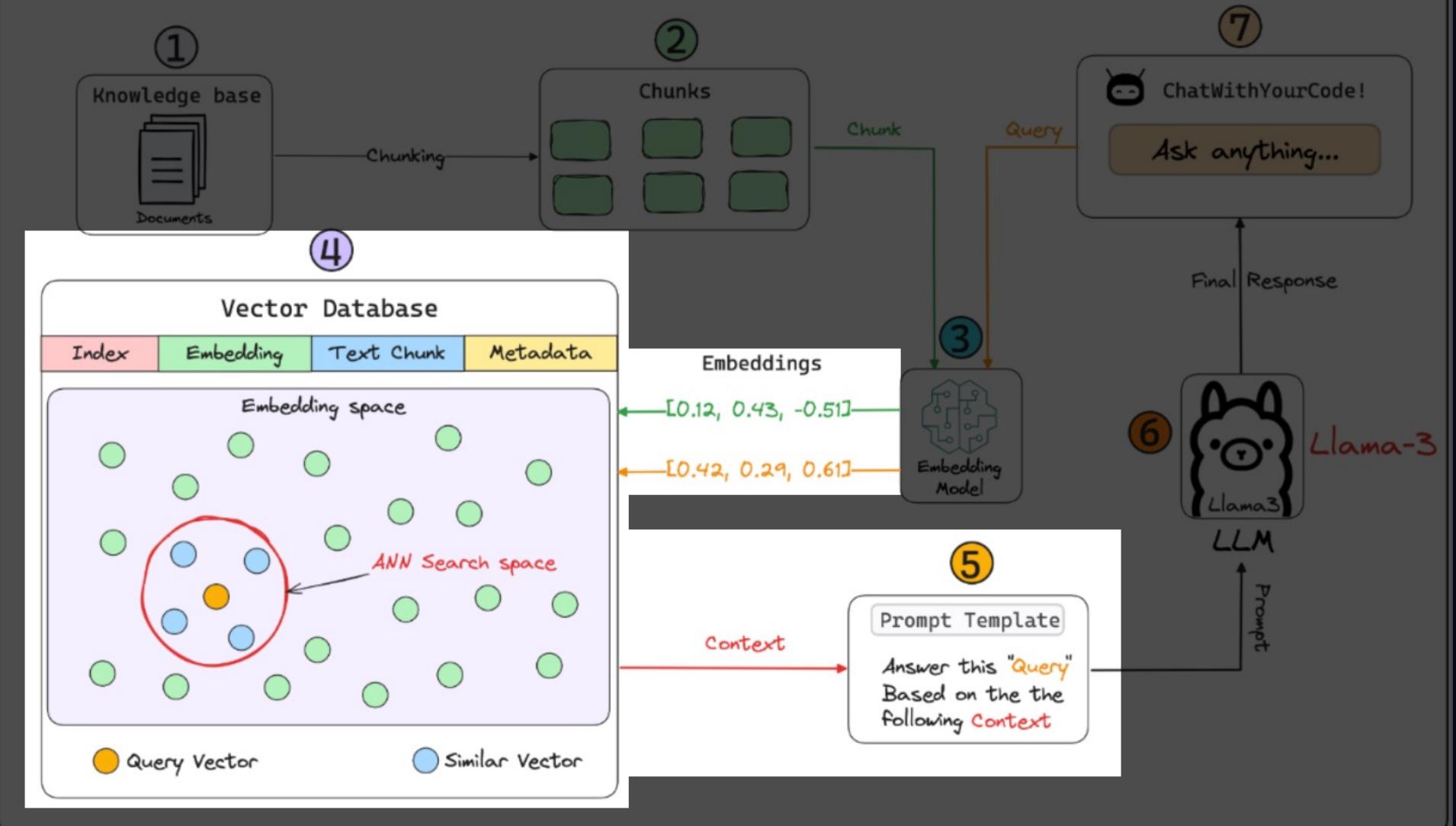
Document Chat RAG using Llama-3!



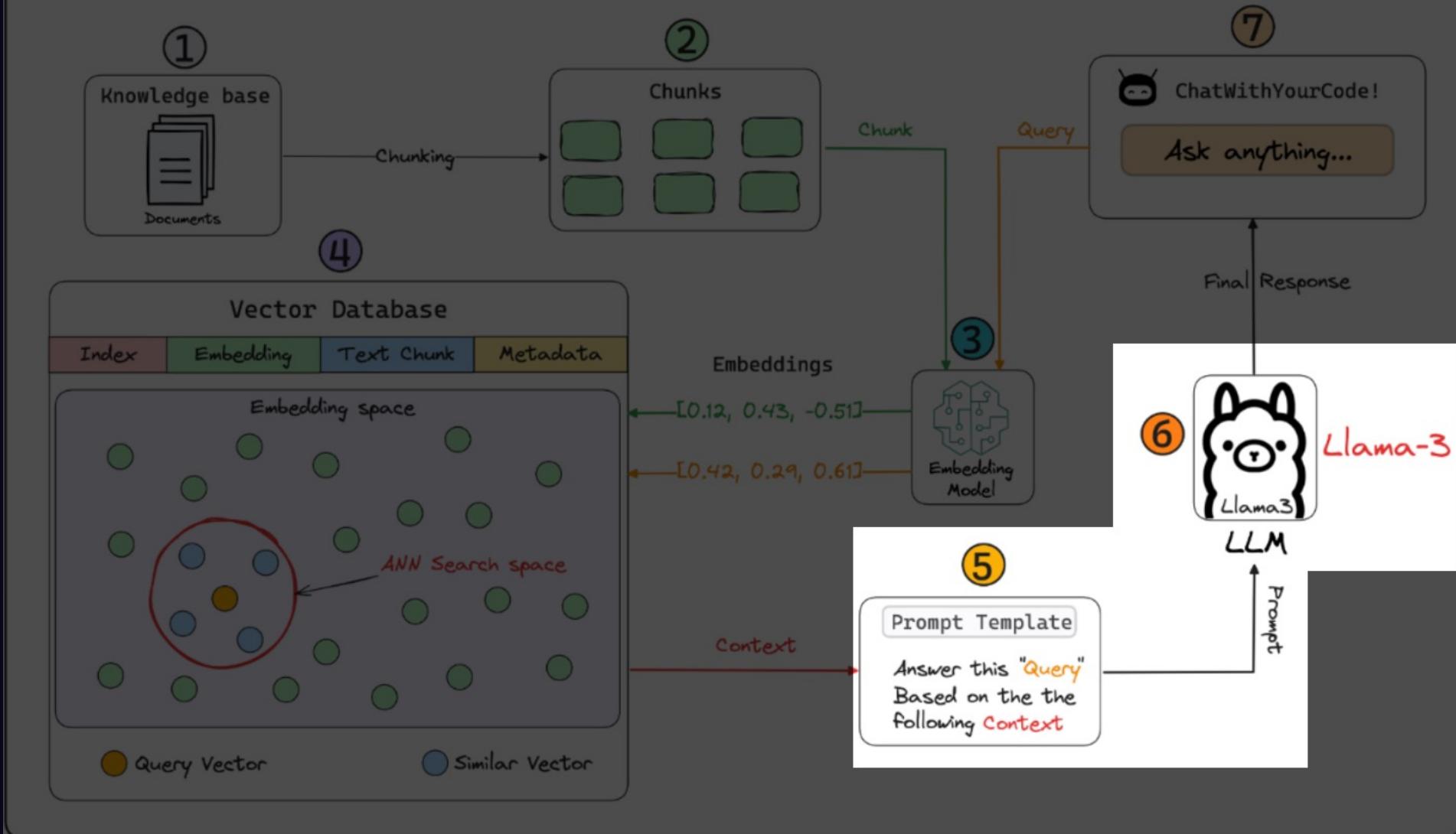
Document Chat RAG using Llama-3!



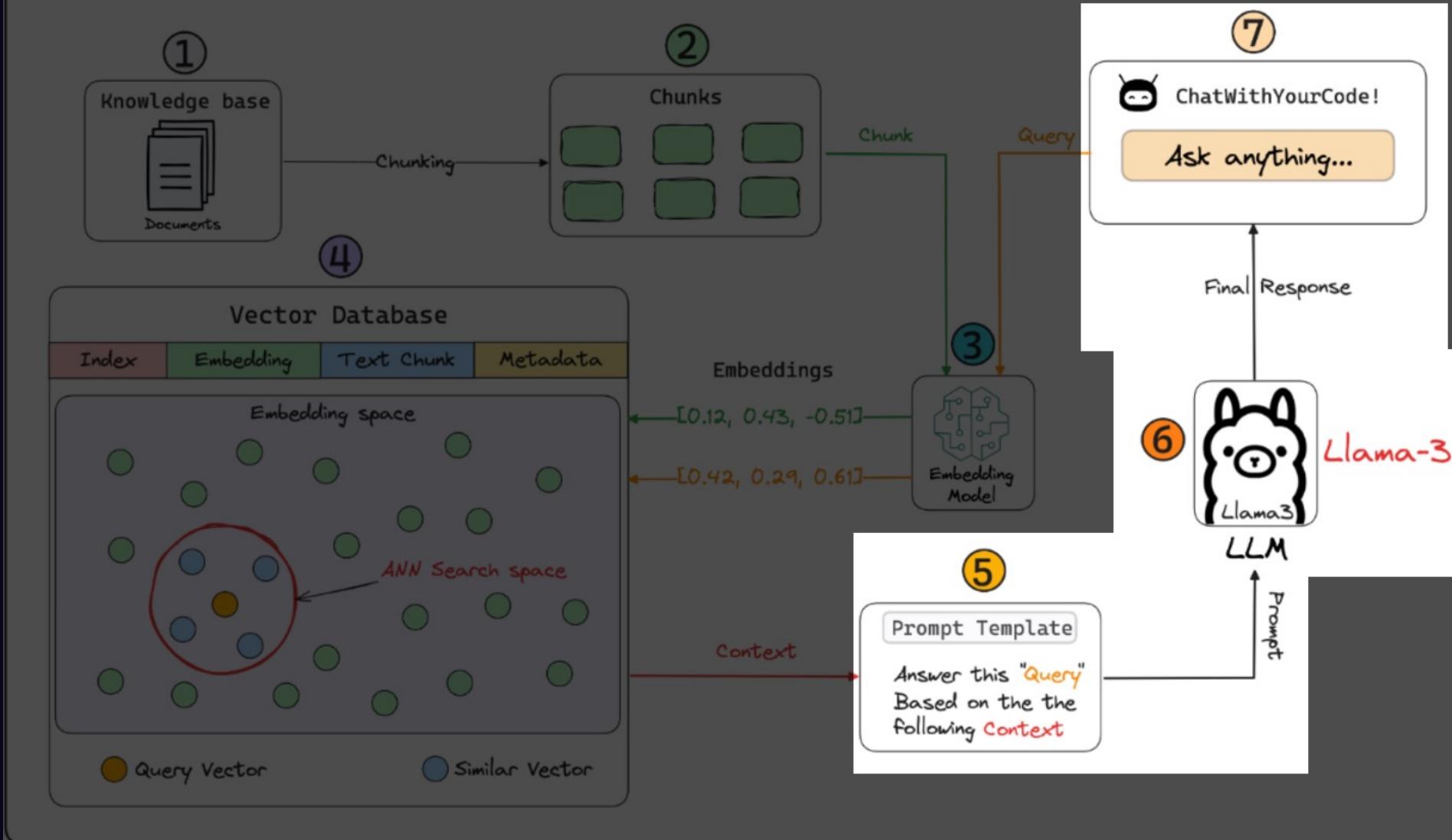
Document Chat RAG using Llama-3!



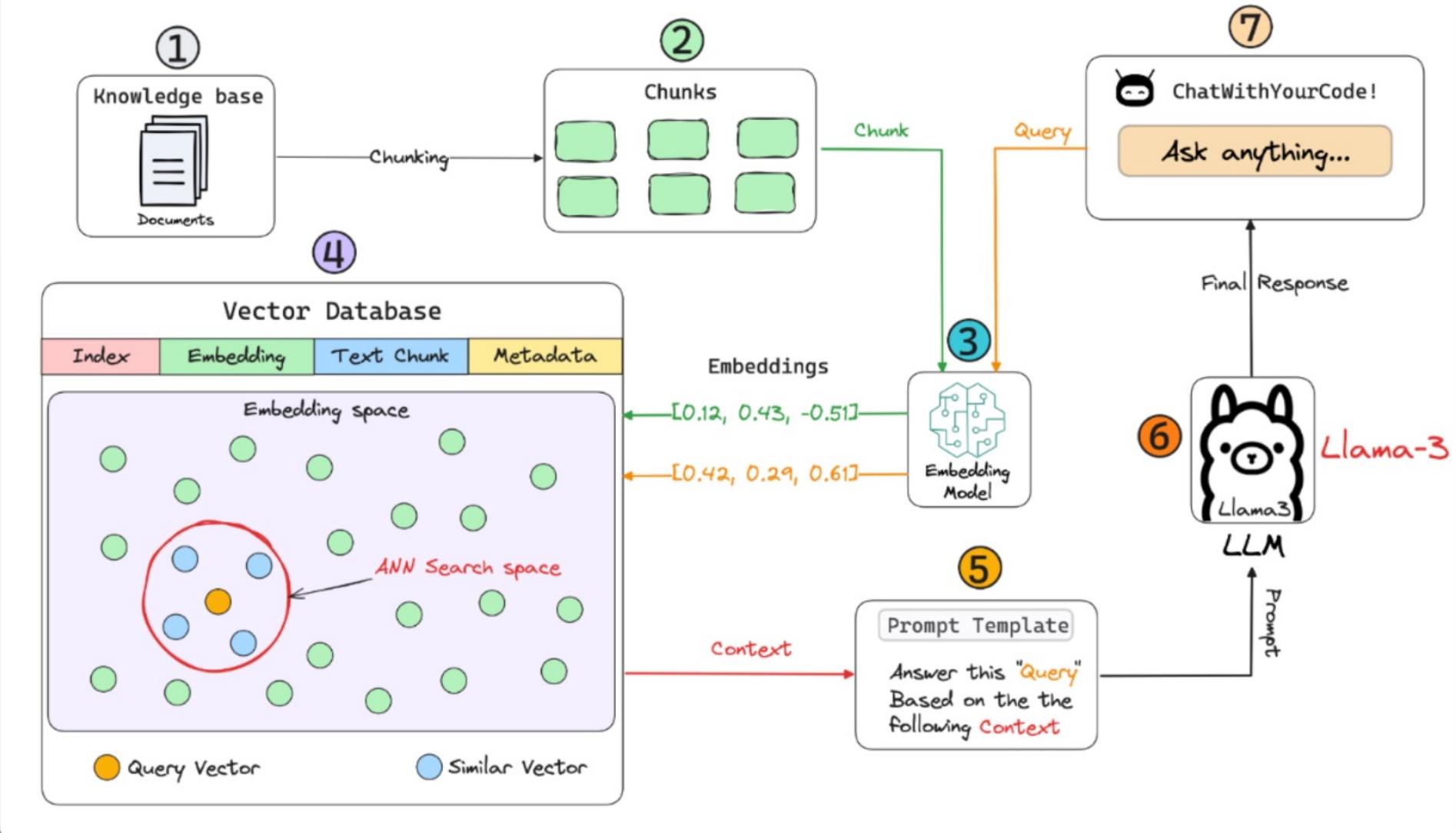
Document Chat RAG using Llama-3!



Document Chat RAG using Llama-3!



Document Chat RAG using Llama-3!



write python code that downloads all the talks from this page:

```
<!DOCTYPE html>
<html>
<head>
  <title>Talks - PyConZA 2024</title>
  ...

```

talks.html

a talk page looks like this

```
<!DOCTYPE html>
<html>
<head>
  <title>Harnessing the Power of Community: Lessons from Speedrunning for the Python Ecosystem and Beyond - PyConZA 2024</title>
  ...

```

talk.html

- ① i need to store all talk data in a csv file with a column for each metadata item



Applying AI with Python

Speaker: [Alex Conway](#).

Track: Data Science

Type: Talk

Room: Lefthand Room / Main Hall

Time: Oct 04 (Fri): 11:30

Duration: 0:45

Artificial Intelligence (AI) is transforming industries and our lives by enabling computers to solve increasingly complex tasks, and Python stands at the forefront of this revolution. In this talk, we will explore how Python's rich ecosystem of AI libraries and frameworks can be harnessed to quickly solve real-world problems without needing a PhD. From computer vision to time series analysis to natural language processing and AI agents, this session will demonstrate practical AI applications in real-world use cases that leverage Python's versatility so you can get started using AI right away.

By the end of the session, you will have a broad overview of AI and of the Python AI landscape together with an understanding of how Python can be used to integrate AI into your projects. The talk will provide practical examples, code snippets, and tips for getting started on building AI-driven solutions. Whether you're a beginner or an experienced developer, this talk will offer valuable insights into applying AI to solve real-world challenges with Python.

Talk Outline:

- Introduction to AI concepts
- Why Python is the go-to language for AI
- Tabular data
- Computer vision
- Time series
- Text Data
- AI Agents
- Conclusion

Key Takeaways:



Jupyter Notebook

Scrape PyConZA2024.ipynb



Talk ID: 1
Speaker ID: 1

--- Talk Data ---

Talk ID: 1
Title: Applying AI with Python
Speaker Name: Alex Conway
Speaker Profile URL: <https://2024.za.pycon.org/users/alxcnwy/>
Talk URL: <https://2024.za.pycon.org/talks/11-applying-ai-with-python/>

Abstract:

Artificial Intelligence (AI) is transforming industries and our lives by enabling computers to solve increasingly complex tasks, and Python stands at the forefront of this revolution. In this talk, we will explore how Python's rich ecosystem of AI libraries and frameworks can be harnessed to quickly solve real-world problems without needing a PhD. From computer vision to time series analysis to natural language processing and AI agents, this session will demonstrate practical AI applications in real-world use cases that leverage Python's versatility so you can get started using AI right away.

By the end of the session, you will have a broad overview of AI and of the Python AI landscape together with an understanding of how Python can be used to integrate AI into your projects. The talk will provide practical examples, code snippets, and tips for getting started on building AI-driven solutions. Whether you're a beginner or an experienced developer, this talk will offer valuable insights into applying AI to solve real-world challenges with Python.

Talk Outline:

- Introduction to AI concepts
- Why Python is the go-to language for AI
- Tabular data
- Computer vision
- Time series
- Text Data
- AI Agents
- Conclusion

Key Takeaways:

- A solid conceptual foundation for how to apply AI
- Knowledge of the Python libraries and tools useful for AI
- Realizing that you don't need to be a math genius to build with AI

--- Speaker Data ---

Photo URL: No photo available
Twitter: <https://twitter.com/alxcnwy>
GitHub: <https://github.com/alxcnwy>
Bio Part 1: Twitter Profile link:<https://twitter.com/alxcnwy>
Bio Part 2: Other Social:<https://www.linkedin.com/in/alxcnwy/>
Bio Part 3: github profile:<https://github.com/alxcnwy>
Bio Part 4: I am a data scientist / hacker with 25 years of coding experience. I've built several startups including my AI development agency NumberBoost and most recently AiSupervision that did Y Combinator and raised several million dollars from VCs. I recently exited that startup and am now working on applied AI via www.numberboost.com
Name: Alex Conway
Profile URL: <https://2024.za.pycon.org/users/alxcnwy/>



Jupyter Notebook

ETL – Load Documents into Database.ipynb



Below is TALK _DATA

Use this talk data below to categorize the talks into topics. return a table with talk id and assigned topic. list the topics as bullet points with a count of talks that fit each topic

TALK_DATA:

Talk ID: 1

Speaker ID: 1

--- Talk Data ---

Talk ID,Title,Abstract,Speaker Name,Speaker Profile URL,Talk

URL,Speaker ID,Photo URL,Twitter,GitHub,Fediverse,Bio

1,Applying AI with Python,"Artificial Intelligence (AI) is transforming industries and our lives by enabling computers to solve increasingly complex tasks, and Python stands at the forefront of this revolution.

In this talk, we will explore how Python's rich ecosystem of AI libraries and frameworks can be harnessed to quickly solve real-world problems without needing a PhD. From computer vision to time series analysis to natural language processing and AI agents, this session will demonstrate practical AI applications in real-world use cases that leverage Python's versatility so you can get started using AI right away.

By the end of the session, you will have a broad overview of AI and of the Python AI landscape together with an understanding of how Python can be used to integrate AI into your projects. The talk will provide practical examples, code snippets, and tips for getting started on building AI-driven solutions. Whether you're a beginner or an experienced developer, this talk will offer valuable insights into

A	B
1	Talk ID Assigned Topic
2	1 AI and Machine Learning
3	2 Community and Tech Education
4	3 Time Series Analysis
5	4 AI and Machine Learning
6	5 Natural Language Processing (NLP)
7	6 Community and Tech Education
8	7 Community and Tech Education
9	8 Community and Tech Education
10	9 Python Libraries and Tools
11	10 Web Development and Frontend
12	11 Data Science and Decision Engines
13	12 Python Libraries and Tools
14	13 Python Libraries and Tools
15	14 Python Libraries and Tools
16	15 Data Quality
17	16 AI and Machine Learning
18	17 Event Streaming and Kafka
19	18 Community and Tech Education
20	19 Python Libraries and Tools
21	20 Robotics
22	21 Open Source and Software Development
23	22 Python Libraries and Tools
24	23 AI and Machine Learning
25	24 Python Libraries and Tools
26	25 IoT and MicroPython
27	26 Community and Tech Education
28	27 Natural Language Processing (NLP)
29	28 Python Libraries and Tools
30	29 Open Source and Software Development
31	30 Web Development and Frontend
32	31 Web Development and Frontend
33	32 Community and Tech Education
34	33 Community and Tech Education
35	34 Open Source and Software Development
36	35 Open Source and Software Development
37	36 Python Libraries and Tools
38	



--- Github ---

speaker_id: 1

github_url: <https://github.com/alxcnwy>

username: alxcnwy

name: Alex Conway

public_repos: 18

total_stars: 104

most_starred_repo_name: Deep-Neural-Networks-for-Video-Classification

most_starred_repo_stars: 44

last_commit_datetime: 2024-10-02T11:26:04+00:00

hours_since_last_commit: 2.1146393786111113

followers: 44

following: 15

created_at: 2014-09-02T18:24:14+00:00

company:

bio: making something people want | ♀ |

ycombinator

W22 | won awards for AI projects from companies like Mercedes-Benz, NTT Japan, etc

location: Cape Town

blog: www.numberboost.com

email:



Jupyter Notebook

ETL – Load Documents into Database.ipynb



```
print(qa_chain.run("How many talks are there about artificial intelligence?"))
```

There are three talks about artificial intelligence. These are:

1. "Applying AI with Python" by Alex Conway
2. "& robotics for all!" by :David Campey
3. "Differentiation Engines: The Elves behind the AI Christmas" by Rajdeep Singh

Wrong answer

Calculation questions are hard for LLMs





Hugging Face

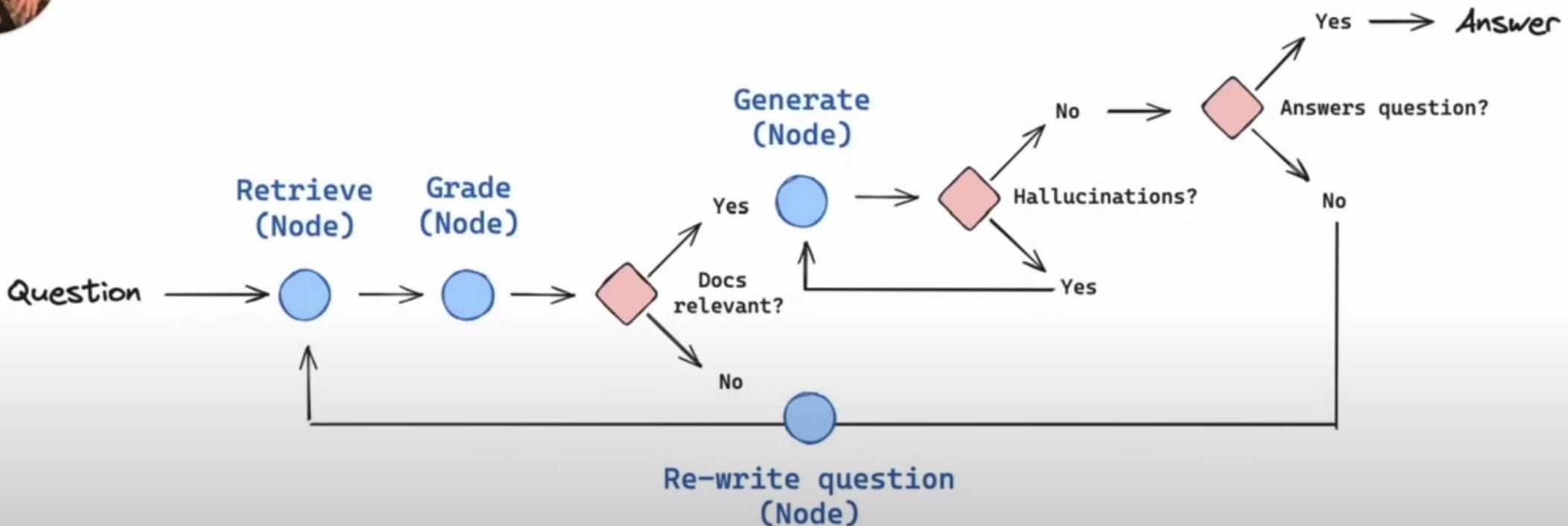
Natural Language Processing

[Text Classification](#)[Token Classification](#)[Table Question Answering](#)[Question Answering](#)[Zero-Shot Classification](#)[Translation](#)[Summarization](#)[Feature Extraction](#)[Text Generation](#)[Text2Text Generation](#)[Fill-Mask](#)[Sentence Similarity](#)

AI Agents



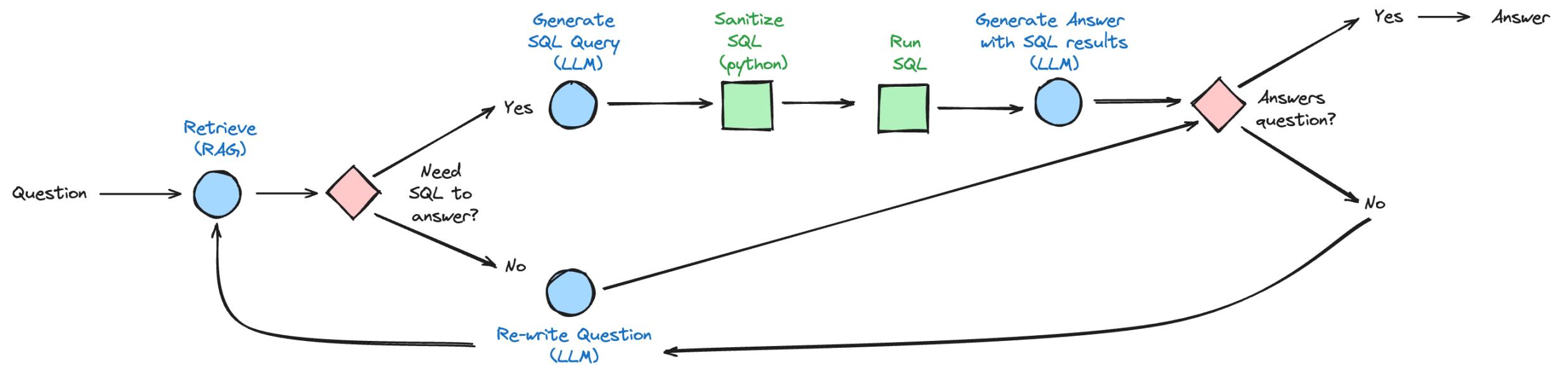
Reasoning: Use reasoning / self-reflection around RAG



```
question = "How many talks are there about artificial intelligence?"
```

Remember the RAG model failed at this...





Homework

Excalidraw Generator

DM me @alxchnwy



Load structured data into SQLite

```
# Normalize column names function
def normalize_column_names(df):
    df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_').str.replace('(', '').str.replace(')', '')
    return df

# Load CSV files into Pandas DataFrames
github_df = pd.read_csv('data/github.csv')
speakers_df = pd.read_csv('data/speakers.csv')
talks_df = pd.read_csv('data/talks.csv')
topics_df = pd.read_csv('data/topics.csv')

# Normalize column names
github_df = normalize_column_names(github_df)
speakers_df = normalize_column_names(speakers_df)
talks_df = normalize_column_names(talks_df)
topics_df = normalize_column_names(topics_df)

# Create a SQLite connection
conn = sqlite3.connect('data/conference_data.db')

# Save the dataframes to SQLite
github_df.to_sql('github', conn, if_exists='replace', index=False)
speakers_df.to_sql('speakers', conn, if_exists='replace', index=False)
talks_df.to_sql('talks', conn, if_exists='replace', index=False)
topics_df.to_sql('topics', conn, if_exists='replace', index=False)
```



	talk_id	title	abstract	speaker_name	speaker_profile_url	talk_url	speaker_id
1	1	Applying AI with Python	Artificial Intelligence (AI) is transforming industries and ...	Alex Conway	https://2024.za.pycon.org/users/alxcnwy/	https://2024.za.pycon.org/talks/...	1
2	2	Harnessing the Power of Community: Lessons from ...	Speedrunning, the art of completing video games as quick...	Nunudzai Mrewa	https://2024.za.pycon.org/users/JustNunuz/	https://2024.za.pycon.org/talks/...	2
3	3	It's About Time: Time-Series Forecasting with Darts	Along with the rise of "AI", data-driven decision-making in ...	Brenden Taylor	https://2024.za.pycon.org/users/brenden.taylor22/	https://2024.za.pycon.org/talks/...	3
4	4	Monitoring and Evaluating LLM Apps with Langfuse	Overview....	Ruan Pretorius	https://2024.za.pycon.org/users/ruankie/	https://2024.za.pycon.org/talks/...	4
5	5	Leveraging the NLTK library for Translation: A Case Study...	In the field of natural language processing (NLP), ...	Alta Saunders	https://2024.za.pycon.org/users/zietsmana/	https://2024.za.pycon.org/talks/...	5
6	6	Creating Personalised Images with Python's Stable Diffusion	Title...	NULL	NULL	https://2024.za.pycon.org/talks/...	6
7	7	Mental Illness and Vulnerability in Tech	In his seminal "Noonday Demon: An Atlas of Depression", ...	Schalk Venter	https://2024.za.pycon.org/users/schalkventer/	https://2024.za.pycon.org/talks/...	6
8	8	Bridging Language Barriers: Making Programming ...	In a diverse and multilingual country like South Africa, th...	Mthetho Sovara	https://2024.za.pycon.org/users/msovara/	https://2024.za.pycon.org/talks/...	7
9	9	Level up your developer experience with Kubernetes	In today's fast-paced development landscape, the efficienc...	Roché Compaan	https://2024.za.pycon.org/users/rochecompaan/	https://2024.za.pycon.org/talks/...	8
10	10	A Novel Way of Creating Multi-Tenant Apps using Django.....	Imagine building a software-as-a-service platform where ...	Shaun De Ponte	https://2024.za.pycon.org/users/nawtybean3d/	https://2024.za.pycon.org/talks/...	9
11	11	Building a Decisioning Engine for Data Scientists: A ...	Machine learning offers a modern replacement for...	Sholto Armstrong	https://2024.za.pycon.org/users/sjnarmstrong/	https://2024.za.pycon.org/talks/...	10
12	12	Implementing an HPy Backend for Cython	HPy (https://hpyproject.org/) is a new API for extending ...	Du Toit Spies	https://2024.za.pycon.org/users/DuToitSpies/	https://2024.za.pycon.org/talks/...	11
13	13	Streamlining HPC Operations: Integrating Django Helpdes...	Django helpdesk is a general purpose ticketing system that...	Binjamin Barsch	https://2024.za.pycon.org/users/bbarsch/	https://2024.za.pycon.org/talks/...	12
14	14	You don't need a data service, you just need an object store...	What are you talking about?...	Gordon Inggs	https://2024.za.pycon.org/users/...	https://2024.za.pycon.org/talks/...	13
15	15	Great Expectations About Data Quality	A story about the City of Cape Town Data Science Branch's...	Ariella Rink	https://2024.za.pycon.org/users/AriellaRink/	https://2024.za.pycon.org/talks/...	14
16	16	Made you look: Using Siamese Neural Networks for Buildin...	What is this talk about?...	Jolanda Becker	https://2024.za.pycon.org/users/jolanda/	https://2024.za.pycon.org/talks/...	15
17	17	Kafka in practice: Lessons learned at Takealot	Overview....	Johan Beyers	https://2024.za.pycon.org/users/johan_beyers/	https://2024.za.pycon.org/talks/...	16
18	18	Power to the people who teach the people to code	Teaching people to code is...tricky....	Sheena O'Connell	https://2024.za.pycon.org/users/sheena/	https://2024.za.pycon.org/talks/...	17
19	19	Adventures in Garbage Collection	Most of the time, garbage collection is an invisible process ...	Bruce Merry	https://2024.za.pycon.org/users/bmerry/	https://2024.za.pycon.org/talks/...	18
20	20	& robotics for all!	Coding & robotics is the new subject in the DBE ...	:David Campey	https://2024.za.pycon.org/users/campey/	https://2024.za.pycon.org/talks/...	19
21	21	Maintaining a PLC communication library called python-...	Python-snap7 is a Python library that enables seamless ...	Gijs Molenaar	https://2024.za.pycon.org/users/gijsm/	https://2024.za.pycon.org/talks/...	20
22	22	Duck, Duck, Python: OLAP Data with DuckDB	➲ DuckDB is an in-process OLAP SQL database that can u...	Luis de Sousa	https://2024.za.pycon.org/users/luisdza/	https://2024.za.pycon.org/talks/...	21
23	23	Differentiation Engines: The Elves behind the AI Christmas	What do all recent advancements in LLMs, Computer Visio...	Rajdeep Singh	https://2024.za.pycon.org/users/singhsegv/	https://2024.za.pycon.org/talks/...	22
24	24	Accelerate your pandas workload using FireDucks at zero ...	In general, a Data Scientist spends significant efforts in ...	Sourav Saha	https://2024.za.pycon.org/users/qsourcev/	https://2024.za.pycon.org/talks/...	23
25	25	Sensor data processing on microcontrollers with ...	Being able to measure physical phenomena is critical to ...	Jon Nordby	https://2024.za.pycon.org/users/jonnor/	https://2024.za.pycon.org/talks/...	24
26	26	Building a Thriving Tech Community	Description of the Talk...	Adeline Makokha	https://2024.za.pycon.org/users/amakokha/	https://2024.za.pycon.org/talks/...	25
27	27	Building a Code Search Engine: Using NLP to Find Similar ...	A developer may decide to implement a library despite the...	Divya Rani	https://2024.za.pycon.org/users/Divya063/	https://2024.za.pycon.org/talks/...	26
28	28	Beyond the Pause: Exploring the Inner Workings of Python...	This talk is designed for Python developers who want to ...	Rupali Kavale	https://2024.za.pycon.org/users/...	https://2024.za.pycon.org/talks/...	27
29	29	Exploring the Adoption & Role of Open Source Software in ...	The adoption of...	Robson Kanhalelo	https://2024.za.pycon.org/users/Kanhalelo/	https://2024.za.pycon.org/talks/...	28
30	30	Using Coding Skills to Make Passive Income	In 2017, I stepped down from my job as CTO of a 150 perso...	Cory Zue	https://2024.za.pycon.org/users/czue/	https://2024.za.pycon.org/talks/...	29

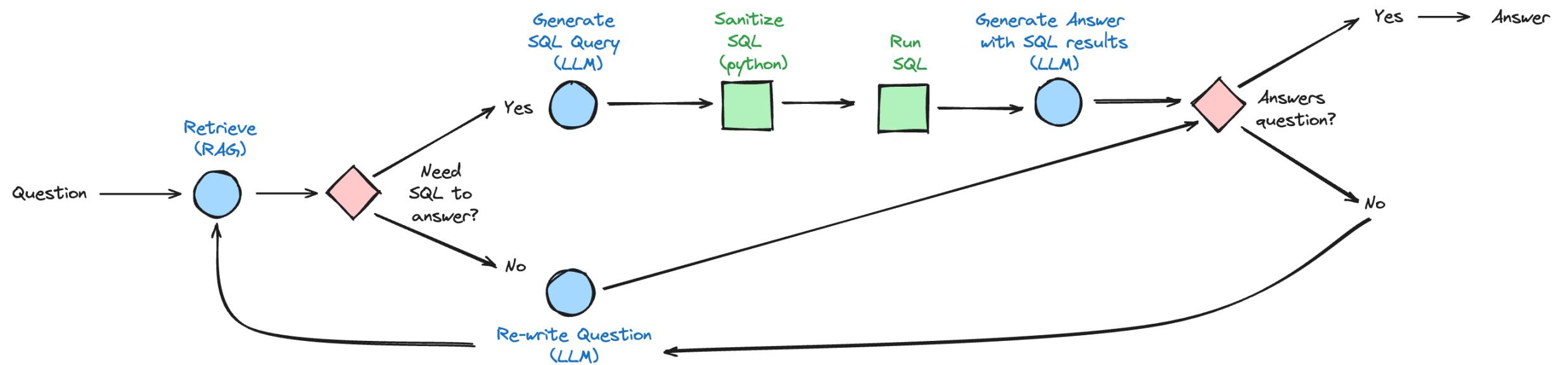
speaker_id	github_url	username	name	public_repos	total_stars	most_starred_repo_name	most_starred_repo_stars	last_commit_d
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	8 https://github.com/sixfeetup	sixfeetup	Six Feet Up	127	210	scaf	79	2024-09-25T17:04:
2	20 https://github.com/gijzelaerr	gijzelaerr	Gijs Molenaar	172	748	python-snap7	648	2024-09-17T16:20:
3	21 https://github.com/luisdza	luisdza	Luis de Sousa	8	3	aerialmzansi-website	3	2024-08-22T13:24:
4	22 https://github.com/singhsegv	singhsegv	Rajdeep	56	51	kitaab_bot	12	2024-09-23T09:53:
5	32 https://github.com/drnlm	drnlm	Neil Muller	30	5	matplotlib-py3	1	2024-09-29T14:11:
6	26 https://github.com/Divya063	Divya063	Divya Rani	190	17	distributedKV	4	2024-08-29T13:34:
7	25 https://github.com/adeline-pepela	adeline-pepela	Adeline Makokha	39	0	NULL	0	2024-09-16T12:50:
8	31 https://github.com/sheenarbw	sheenarbw	sheenarbw	76	22	tutorial-airflow	6	2024-10-02T13:00:
9	17 https://github.com/sheenarbw	sheenarbw	sheenarbw	76	22	tutorial-airflow	6	2024-10-02T13:00:
10	29 https://github.com/czue	czue	Cory Zue	67	1016	celery-progress	464	2024-09-26T13:58:
11	30 https://github.com/sheenarbw	sheenarbw	sheenarbw	76	22	tutorial-airflow	6	2024-09-23T12:49:
12	24 https://github.com/emlearn	emlearn	NULL	2	546	emlearn	483	2024-10-01T00:07:
13	13 https://github.com/cityofcapetown	cityofcapetown	City of Cape Town	44	20	loadshedding-signal-bot	4	2024-09-28T11:56:
14	27 https://github.com/coderquill	coderquill	Rupali Kavale	75	3	topstar	3	2024-09-23T00:53:
15	28 https://github.com/Kanhalelor	Kanhalelor	Robson	54	2	CSS-Flexbox-Layout	1	2024-07-12T12:31:
16	1 https://github.com/alxcnwy	alxcnwy	Alex Conway	18	104	Deep-Neural-Networks-for-Video-Classification	44	2024-10-02T11:26:
17	16 https://github.com/jbeyers	jbeyers	Johan Beyers	42	57	django-prelaunch	36	2024-08-14T04:57:
18	2 https://github.com/JustNunuz	JustNunuz	Nunudzai Mrewa	10	0	NULL	0	2024-09-13T13:33:
19	11 https://github.com/DuToitSpies	DuToitSpies	Du Toit Spies	11	1	cython	1	2024-08-10T18:13:
20	12 https://github.com/kode2go	kode2go	kode2go	74	7	nithecs	2	2024-06-01T22:11:
21	6 https://github.com/schalkventer	schalkventer	Schalk Venter	171	165	schalkventer.me	17	2024-09-29T08:11:
22	10 https://github.com/sjnarmstrong	sjnarmstrong	Sholto Armstrong	27	131	gray-code-structured-light	112	2024-04-09T17:40:
23	9 https://github.com/nawtybean	nawtybean	Shaun De Ponte	16	4	KickstarterPrediction	3	2024-09-25T11:53:
24	4 https://github.com/ruankie	ruankie	Ruan Pretorius	71	112	ecrivai	68	2024-10-02T00:42:

Jupyter Notebook

Let's make RAG better by giving the AI tools
and giving it self-reflection and reasoning ability

Agent.ipynb





```
def agent(question, max_retries=2):
    attempt = 0
    answers_question = "no"
    while answers_question.lower() != "yes" and attempt < max_retries:
        attempt += 1
        if attempt > 1:
            print(f"> Rewriting question {attempt}: {question}")
            clear_notebook_output()
        # figure out if SQL or LLM question
        prompt_decide_LLM_or_SQL = get_prompt_decide_LLM_or_SQL(question)
        question_type = run_chat_query(prompt_decide_LLM_or_SQL)
        # reasoning
        if question_type == "SQL":
            print("Using SQL to inform answer")
            # write sql query
            prompt_write_SQL_query_for_question = get_prompt_write_SQL_query_for_question(question)
            sql_query = run_chat_query(prompt_write_SQL_query_for_question)

            # query sql database
            sql_query = sanitize_sql_query(sql_query)
            print(f"> Running SQL Query: {sql_query}")
            sql_results = run_sql_query(sql_query)

            # construct answer with insights from the data
            prompt_analyze_sql = get_prompt_analyze_sql(question, sql_results)
            answer = print_markdown(run_chat_query(prompt_analyze_sql))
        else:
            print("Using RAG to inform answer")
            # use RAG to answer
            prompt_analyze_llm = get_prompt_analyze_llm(question)
            answer = print_markdown(run_chat_query(prompt_analyze_llm))

        # check if answer actually answers the question
        prompt_analyze_llm = get_prompt_check_answer(question, answer)
        answers_question = str(run_chat_query(prompt_analyze_llm))

        # rewrite question
        prompt_analyze_llm = get_prompt_rewrite_question(question)
        question = run_chat_query(prompt_analyze_llm)

    return answer
```



```
def agent(question, max_retries=2):
    attempt = 0
    answers_question = "no"
    while answers_question.lower() != "yes" and attempt < max_retries:
        attempt += 1
        if attempt > 1:
            print(f"> Rewriting question {attempt}: {question}")
            clear_notebook_output()
        # figure out if SQL or LLM question
        prompt_decide_LLM_or_SQL = get_prompt_decide_LLM_or_SQL(question)
        question_type = run_chat_query(prompt_decide_LLM_or_SQL)
        # reasoning
        if question_type == "SQL":
            print("Using SQL to inform answer")
            # write sql query
            prompt_write_SQL_query_for_question = get_prompt_write_SQL_query_for_question(question)
            sql_query = run_chat_query(prompt_write_SQL_query_for_question)

            # query sql database
            sql_query = sanitize_sql_query(sql_query)
            print(f"> Running SQL Query: {sql_query}")
            sql_results = run_sql_query(sql_query)

            # construct answer with insights from the data
            prompt_analyze_sql = get_prompt_analyze_sql(question, sql_results)
            answer = print_markdown(run_chat_query(prompt_analyze_sql))
        else:
            print("Using RAG to inform answer")
            # use RAG to answer
            prompt_analyze_llm = get_prompt_analyze_llm(question)
            answer = print_markdown(run_chat_query(prompt_analyze_llm))

        # check if answer actually answers the question
        prompt_analyze_llm = get_prompt_check_answer(question, answer)
        answers_question = str(run_chat_query(prompt_analyze_llm))

    # rewrite question
    prompt_analyze_llm = get_prompt_rewrite_question(question)
    question = run_chat_query(prompt_analyze_llm)

return answer
```



```
def agent(question, max_retries=2):
    attempt = 0
    answers_question = "no"
    while answers_question.lower() != "yes" and attempt < max_retries:
        attempt += 1
        if attempt > 1:
            print(f"> Rewriting question {attempt}: {question}")
            clear_notebook_output()
        # figure out if SQL or LLM question
        prompt_decide_LLM_or_SQL = get_prompt_decide_LLM_or_SQL(question)
        question_type = run_chat_query(prompt_decide_LLM_or_SQL)
        # reasoning
        if question_type == "SQL":
            print("Using SQL to inform answer")
            # write sql query
            prompt_write_SQL_query_for_question = get_prompt_write_SQL_query_for_question(question)
            sql_query = run_chat_query(prompt_write_SQL_query_for_question)

            # query sql database
            sql_query = sanitize_sql_query(sql_query)
            print(f"> Running SQL Query: {sql_query}")
            sql_results = run_sql_query(sql_query)

            # construct answer with insights from the data
            prompt_analyze_sql = get_prompt_analyze_sql(question, sql_results)
            answer = print_markdown(run_chat_query(prompt_analyze_sql))
        else:
            print("Using RAG to inform answer")
            # use RAG to answer
            prompt_analyze_llm = get_prompt_analyze_llm(question)
            answer = print_markdown(run_chat_query(prompt_analyze_llm))

        # check if answer actually answers the question
        prompt_analyze_llm = get_prompt_check_answer(question, answer)
        answers_question = str(run_chat_query(prompt_analyze_llm))

        # rewrite question
        prompt_analyze_llm = get_prompt_rewrite_question(question)
        question = run_chat_query(prompt_analyze_llm)

    return answer
```



```
def agent(question, max_retries=2):
    attempt = 0
    answers_question = "no"
    while answers_question.lower() != "yes" and attempt < max_retries:
        attempt += 1
        if attempt > 1:
            print(f"> Rewriting question {attempt}: {question}")
            clear_notebook_output()
    # figure out if SQL or LLM question
    prompt_decide_LLM_or_SQL = get_prompt_decide_LLM_or_SQL(question)
    question_type = run_chat_query(prompt_decide_LLM_or_SQL)
    # reasoning
    if question_type == "SQL":
        print("Using SQL to inform answer")
        # write sql query
        prompt_write_SQL_query_for_question = get_prompt_write_SQL_query_for_question(question)
        sql_query = run_chat_query(prompt_write_SQL_query_for_question)

        # query sql database
        sql_query = sanitize_sql_query(sql_query)
        print(f"> Running SQL Query: {sql_query}")
        sql_results = run_sql_query(sql_query)

        # construct answer with insights from the data
        prompt_analyze_sql = get_prompt_analyze_sql(question, sql_results)
        answer = print_markdown(run_chat_query(prompt_analyze_sql))
    else:
        print("Using RAG to inform answer")
        # use RAG to answer
        prompt_analyze_llm = get_prompt_analyze_llm(question)
        answer = print_markdown(run_chat_query(prompt_analyze_llm))

    # check if answer actually answers the question
    prompt_analyze_llm = get_prompt_check_answer(question, answer)
    answers_question = str(run_chat_query(prompt_analyze_llm))

    # rewrite question
    prompt_analyze_llm = get_prompt_rewrite_question(question)
    question = run_chat_query(prompt_analyze_llm)

return answer
```



```
def agent(question, max_retries=2):
    attempt = 0
    answers_question = "no"
    while answers_question.lower() != "yes" and attempt < max_retries:
        attempt += 1
        if attempt > 1:
            print(f"> Rewriting question {attempt}: {question}")
            clear_notebook_output()
    # figure out if SQL or LLM question
    prompt_decide_LLM_or_SQL = get_prompt_decide_LLM_or_SQL(question)
    question_type = run_chat_query(prompt_decide_LLM_or_SQL)
    # reasoning
    if question_type == "SQL":
        print("Using SQL to inform answer")
        # write sql query
        prompt_write_SQL_query_for_question = get_prompt_write_SQL_query_for_question(question)
        sql_query = run_chat_query(prompt_write_SQL_query_for_question)

        # query sql database
        sql_query = sanitize_sql_query(sql_query)
        print(f"> Running SQL Query: {sql_query}")
        sql_results = run_sql_query(sql_query)

        # construct answer with insights from the data
        prompt_analyze_sql = get_prompt_analyze_sql(question, sql_results)
        answer = print_markdown(run_chat_query(prompt_analyze_sql))
    else:
        print("Using RAG to inform answer")
        # use RAG to answer
        prompt_analyze_llm = get_prompt_analyze_llm(question)
        answer = print_markdown(run_chat_query(prompt_analyze_llm))

    # check if answer actually answers the question
    prompt_analyze_llm = get_prompt_check_answer(question, answer)
    answers_question = str(run_chat_query(prompt_analyze_llm))

    # rewrite question
    prompt_analyze_llm = get_prompt_rewrite_question(question)
    question = run_chat_query(prompt_analyze_llm)

return answer
```



```
def agent(question, max_retries=2):
    attempt = 0
    answers_question = "no"
    while answers_question.lower() != "yes" and attempt < max_retries:
        attempt += 1
        if attempt > 1:
            print(f"> Rewriting question {attempt}: {question}")
            clear_notebook_output()
        # figure out if SQL or LLM question
        prompt_decide_LLM_or_SQL = get_prompt_decide_LLM_or_SQL(question)
        question_type = run_chat_query(prompt_decide_LLM_or_SQL)
        # reasoning
        if question_type == "SQL":
            print("Using SQL to inform answer")
            # write sql query
            prompt_write_SQL_query_for_question = get_prompt_write_SQL_query_for_question(question)
            sql_query = run_chat_query(prompt_write_SQL_query_for_question)

            # query sql database
            sql_query = sanitize_sql_query(sql_query)
            print(f"> Running SQL Query: {sql_query}")
            sql_results = run_sql_query(sql_query)

            # construct answer with insights from the data
            prompt_analyze_sql = get_prompt_analyze_sql(question, sql_results)
            answer = print_markdown(run_chat_query(prompt_analyze_sql))
        else:
            print("Using RAG to inform answer")
            # use RAG to answer
            prompt_analyze_llm = get_prompt_analyze_llm(question)
            answer = print_markdown(run_chat_query(prompt_analyze_llm))

        # check if answer actually answers the question
        prompt_analyze_llm = get_prompt_check_answer(question, answer)
        answers_question = str(run_chat_query(prompt_analyze_llm))

        # rewrite question
        prompt_analyze_llm = get_prompt_rewrite_question(question)
        question = run_chat_query(prompt_analyze_llm)

    return answer
```



```
def agent(question, max_retries=2):
    attempt = 0
    answers_question = "no"
    while answers_question.lower() != "yes" and attempt < max_retries:
        attempt += 1
        if attempt > 1:
            print(f"> Rewriting question {attempt}: {question}")
            clear_notebook_output()
        # figure out if SQL or LLM question
        prompt_decide_LLM_or_SQL = get_prompt_decide_LLM_or_SQL(question)
        question_type = run_chat_query(prompt_decide_LLM_or_SQL)
        # reasoning
        if question_type == "SQL":
            print("Using SQL to inform answer")
            # write sql query
            prompt_write_SQL_query_for_question = get_prompt_write_SQL_query_for_question(question)
            sql_query = run_chat_query(prompt_write_SQL_query_for_question)

            # query sql database
            sql_query = sanitize_sql_query(sql_query)
            print(f"> Running SQL Query: {sql_query}")
            sql_results = run_sql_query(sql_query)

            # construct answer with insights from the data
            prompt_analyze_sql = get_prompt_analyze_sql(question, sql_results)
            answer = print_markdown(run_chat_query(prompt_analyze_sql))
        else:
            print("Using RAG to inform answer")
            # use RAG to answer
            prompt_analyze_llm = get_prompt_analyze_llm(question)
            answer = print_markdown(run_chat_query(prompt_analyze_llm))

        # check if answer actually answers the question
        prompt_analyze_llm = get_prompt_check_answer(question, answer)
        answers_question = str(run_chat_query(prompt_analyze_llm))

        # rewrite question
        prompt_analyze_llm = get_prompt_rewrite_question(question)
        question = run_chat_query(prompt_analyze_llm)

    return answer
```



```
def agent(question, max_retries=2):
    attempt = 0
    answers_question = "no"
    while answers_question.lower() != "yes" and attempt < max_retries:
        attempt += 1
        if attempt > 1:
            print(f"> Rewriting question {attempt}: {question}")
            clear_notebook_output()
        # figure out if SQL or LLM question
        prompt_decide_LLM_or_SQL = get_prompt_decide_LLM_or_SQL(question)
        question_type = run_chat_query(prompt_decide_LLM_or_SQL)
        # reasoning
        if question_type == "SQL":
            print("Using SQL to inform answer")
            # write sql query
            prompt_write_SQL_query_for_question = get_prompt_write_SQL_query_for_question(question)
            sql_query = run_chat_query(prompt_write_SQL_query_for_question)

            # query sql database
            sql_query = sanitize_sql_query(sql_query)
            print(f"> Running SQL Query: {sql_query}")
            sql_results = run_sql_query(sql_query)

            # construct answer with insights from the data
            prompt_analyze_sql = get_prompt_analyze_sql(question, sql_results)
            answer = print_markdown(run_chat_query(prompt_analyze_sql))
        else:
            print("Using RAG to inform answer")
            # use RAG to answer
            prompt_analyze_llm = get_prompt_analyze_llm(question)
            answer = print_markdown(run_chat_query(prompt_analyze_llm))

        # check if answer actually answers the question
        prompt_analyze_llm = get_prompt_check_answer(question, answer)
        answers_question = str(run_chat_query(prompt_analyze_llm))

        # rewrite question
        prompt_analyze_llm = get_prompt_rewrite_question(question)
        question = run_chat_query(prompt_analyze_llm)

    return answer
```



```
print(qa_chain.run("How many talks are there about artificial intelligence?"))
```

There are three talks about artificial intelligence. These are:

1. "Applying AI with Python" by Alex Conway
2. "& robotics for all!" by :David Campey
3. "Differentiation Engines: The Elves behind the AI Christmas" by Rajdeep Singh

RAG = Wrong!



```
agent("How many talks are there about artificial intelligence?")
```

Using SQL to inform answer

- There are 4 distinct talks about artificial intelligence at the conference.

```
COUNT(DISTINCT t.talk_id)
```

4

Insights

- With 4 talks focused on artificial intelligence, it suggests a strong interest in this field at the conference.
- The diversity of AI topics could provide a comprehensive overview for attendees interested in various AI applications and techniques.
- This indicates that AI is a significant area of focus and likely a key theme of the conference.

Agent = Correct!





Alex Conway .eth @alxcnwy

like this post if you know wtf is a LAM



Conclusion



```
def ai_model(input):  
    #  
    # algorithms f(input) = output  
    #  
    return output
```



```
def ai_model(input):  
    #  
    #  
    #  
    return output
```



ChatGPT

include
buzzwords



```
def ai_model(input):  
    #  
    #  
    #  
    return output
```



Thank You!

Alex Conway

alex@numberboost.com

x.com/alxchnwy