

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



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](#)

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.)
- And more...



```
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

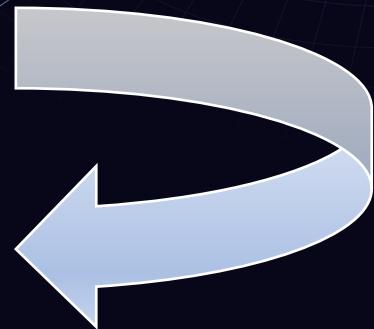


Image / Video



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

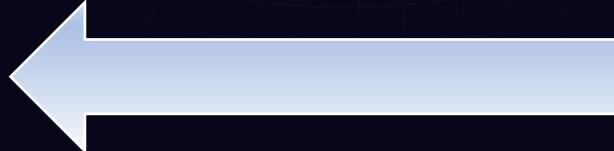


Image / Video



Audio



	A	B	C	D
1	Experience_Years	Age	Postgrad	Salary
2		5	28	FALSE
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

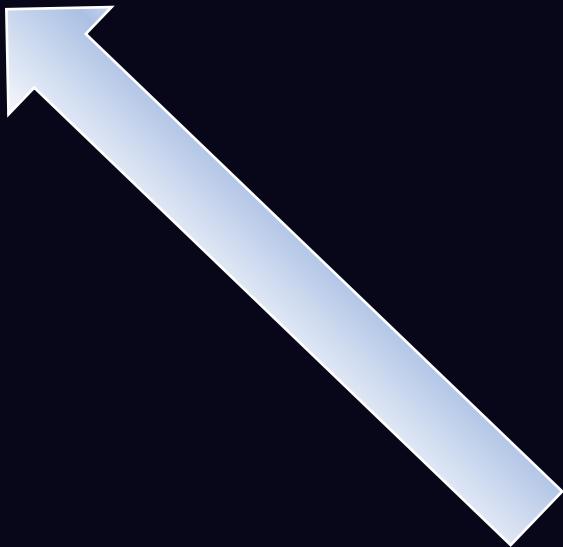


Image / Video



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
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

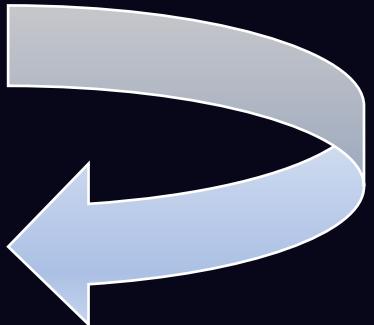


	A	B	C	D
1	Experience_Years	Age	Postgrad	Salary
2		5	28	FALSE
3		1	21	TRUE
4		3	23	FALSE
5		2	22	TRUE
6		1	17	TRUE
7		25	62	TRUE
8		19	54	FALSE
9		2	21	FALSE
10		10	36	FALSE
				61500

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

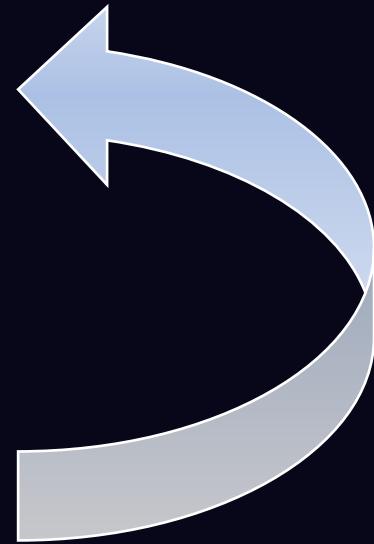


Image / Video



Audio



Why Python for AI?

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

 TensorFlow



 PyTorch



 Keras

 XGBoost

 statsmodels

 Hugging Face

 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

[Image-Text-to-Text](#) [Visual Question Answering](#)
[Document Question Answering](#) [Video-Text-to-Text](#)
[Any-to-Any](#)

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](#)

Reinforcement Learning

[Reinforcement Learning](#) [Robotics](#)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[black-forest-labs/FLUX.1-dev](#)[Text-to-Image](#) Updated Aug 16 ↓ 1.12M ⚡ 5.02k[meta-llama/Llama-3.2-1B](#)[Text Generation](#) Updated 3 days ago ↓ 47.4k ⚡ 292[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[meta-llama/Llama-3.2-3B-Instruct](#)[Text Generation](#) Updated 8 days ago ↓ 69.7k ⚡ 241[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[meta-llama/Llama-3.2-90B-Vision-Instruct](#)[Image-Text-to-Text](#) Updated 3 days ago ↓ 11.9k ⚡ 162[ostris/OpenFLUX.1](#)[Text-to-Image](#) Updated 2 days ago ↓ 142 ⚡ 167[vidore/colqwen2-v0.1](#)

Updated 1 day ago ↓ 3.01k ⚡ 70

[black-forest-labs/FLUX.1-schnell](#)[Text-to-Image](#) Updated Aug 16 ↓ 939k ⚡ 2.44k[glif/how2draw](#)[Text-to-Image](#) Updated 10 days ago ↓ 4.75k ⚡ 141[allenai/MolmoE-1B-0924](#)[Image-Text-to-Text](#) Updated 2 days ago ↓ 4.64k ⚡ 85[openai/whisper-large-v3-turbo](#)[Automatic Speech Recognition](#) Updated about 22 hours ago ↓ 5.73k ⚡ 263[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[openbmb/MiniCPM-Embedding](#)[Feature Extraction](#) Updated 1 day ago ↓ 116k ⚡ 199[allenai/Molmo-72B-0924](#)[Image-Text-to-Text](#) Updated 1 day ago ↓ 1.63k ⚡ 196[meta-llama/Llama-3.2-3B](#)[Text Generation](#) Updated 6 days ago ↓ 22.6k ⚡ 135[OnomaAIResearch/Illustrious-xl-early-release-v0](#)[Text-to-Image](#) Updated 2 days ago ↓ 3.56k ⚡ 160[amd/AMD-Llama-135m](#)[Text Generation](#) Updated Aug 6 ↓ 2.77k ⚡ 89[allenai/Molmo-7B-0-0924](#)[Image-Text-to-Text](#) Updated 1 day ago ↓ 1.96k ⚡ 112[meta-llama/Llama-3.1-8B-Instruct](#)[Text Generation](#) Updated 8 days ago ↓ 3.23M ⚡ 2.69k[nvidia/Llama-3_1-Nemotron-51B-Instruct](#)[Text Generation](#) Updated 3 days ago ↓ 13.6k ⚡ 164[openai/whisper-large-v3](#)[Automatic Speech Recognition](#) Updated Aug 12 ↓ 4.14M ⚡ 3.5k[AIDC-AI/Ovis1.6-Gemma2-9B](#)[Image-Text-to-Text](#) Updated 7 days ago ↓ 6.58k ⚡ 184



[Models](#) [Datasets](#) [Spaces](#) [Posts](#) [Docs](#) [Solutions](#) [Pricing](#) [Log In](#) [Sign Up](#)

Image/ Video

Text

Audio

Table

[Tasks](#) [Libraries](#) [Datasets](#) [Languages](#) [Licenses](#) [Other](#)

Multimodal
 [Image-Text-to-Text](#) [Visual Question Answering](#)
 [Document Question Answering](#) [Video-Text-to-Text](#)
 [Any-to-Any](#)
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](#)
Reinforcement Learning
 [Reinforcement Learning](#) [Robotics](#)
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

 [black-forest-labs/FLUX.1-dev](#)
 Text-to-Image • Updated Aug 16 • ↓ 1.12M • ⚡ • 5.02k

 [meta-llama/Llama-3.2-1B](#)
 Text Generation • Updated 3 days ago • ↓ 47.4k • 292

 [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

 [meta-llama/Llama-3.2-3B-Instruct](#)
 Text Generation • Updated 8 days ago • ↓ 69.7k • 241

 [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

 [meta-llama/Llama-3.2-90B-Vision-Instruct](#)
 Image-Text-to-Text • Updated 3 days ago • ↓ 11.9k • 162

 [ostris/OpenFLUX.1](#)
 Text-to-Image • Updated 2 days ago • ↓ 142 • 167

 [vidore/colqwen2-v0.1](#)
Updated 1 day ago • ↓ 3.01k • 70

 [black-forest-labs/FLUX.1-schnell](#)
 Text-to-Image • Updated Aug 16 • ↓ 939k • ⚡ • 2.44k

 [glif/how2draw](#)
 Text-to-Image • Updated 10 days ago • ↓ 4.75k • ⚡ • 141

 [allenai/MolmoE-1B-0924](#)
 Image-Text-to-Text • Updated 2 days ago • ↓ 4.64k • 85

 [openai/whisper-large-v3-turbo](#)
 Automatic Speech Recognition • Updated about 22 hours ago • ↓ 5.73k • 263

 [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

 [openbmb/MiniCPM-Embedding](#)
 Feature Extraction • Updated 1 day ago • ↓ 116k • 199

 [allenai/Molmo-72B-0924](#)
 Image-Text-to-Text • Updated 1 day ago • ↓ 1.63k • 196

 [meta-llama/Llama-3.2-3B](#)
 Text Generation • Updated 6 days ago • ↓ 22.6k • ⚡ • 135

 [OnomaAIResearch/Illustrious-xl-early-release-v0](#)
 Text-to-Image • Updated 2 days ago • ↓ 3.56k • ⚡ • 160

 [amd/AMD-Llama-135m](#)
 Text Generation • Updated Aug 6 • ↓ 2.77k • 89

 [allenai/Molmo-7B-0-0924](#)
 Image-Text-to-Text • Updated 1 day ago • ↓ 1.96k • 112

 [meta-llama/Llama-3.1-8B-Instruct](#)
 Text Generation • Updated 8 days ago • ↓ 3.23M • ⚡ • 2.69k

 [nvidia/Llama-3_1-Nemotron-51B-Instruct](#)
 Text Generation • Updated 3 days ago • ↓ 13.6k • 164

 [openai/whisper-large-v3](#)
 Automatic Speech Recognition • Updated Aug 12 • ↓ 4.14M • ⚡ • 3.5k

 [AIDC-AI/Ovis1.6-Gemma2-9B](#)
 Image-Text-to-Text • Updated 7 days ago • ↓ 6.58k • 184

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

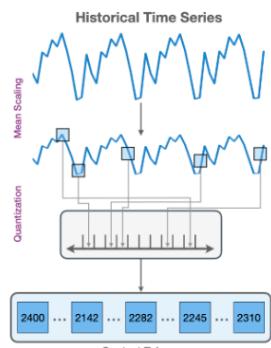
Use this model

Chronos-T5 (Tiny)

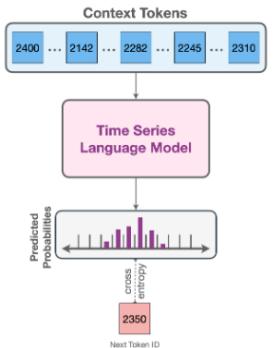
Chronos is a family of **pretrained time series forecasting models** based on language model architectures. A time series is transformed into a sequence of tokens via scaling and quantization, and a language model is trained on these tokens using the cross-entropy loss. Once trained, probabilistic forecasts 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](#).

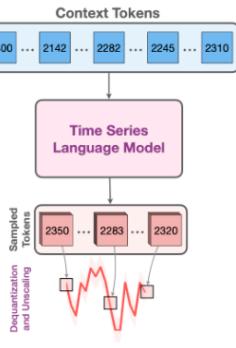
Time Series Tokenization



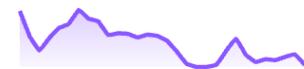
Training



Inference



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 for... • 8 items • Updated Jun 27 • △ 28



aws amazon/chronos-t5-tiny

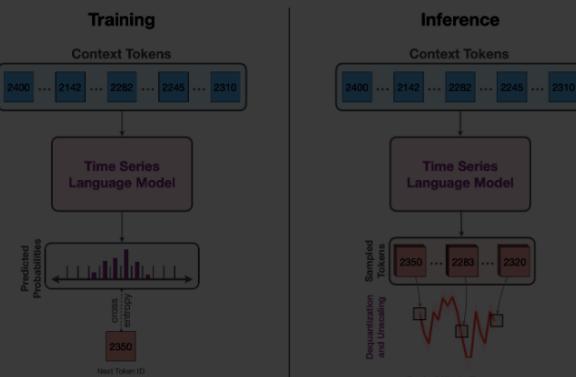
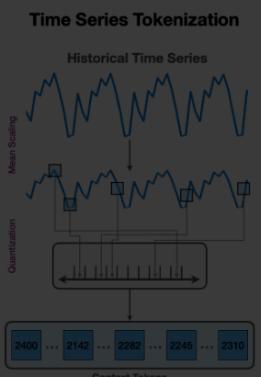
[Time Series Forecasting](#) [Transformers](#)
[text-generation-inference](#) [Inference Endpoints](#)[Model card](#) [Files and versions](#)

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 from the language model.

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](#).



Collection including amazon/chronos-t5-tiny

Chronos Models & Datasets

Chronos: Pretrained (language) models for time series forecasting

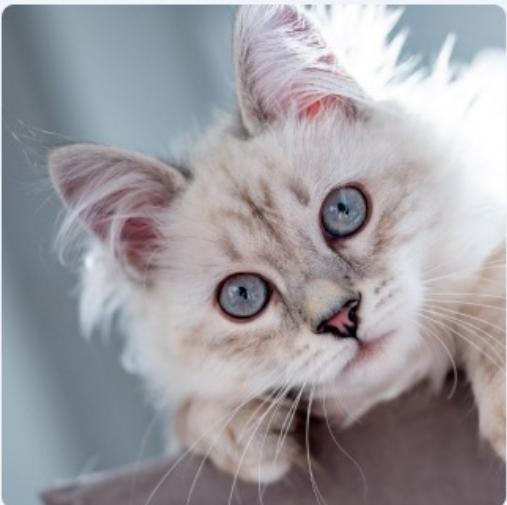
8 items · Updated Jun 27 · 28



Image / Video Data



Classification

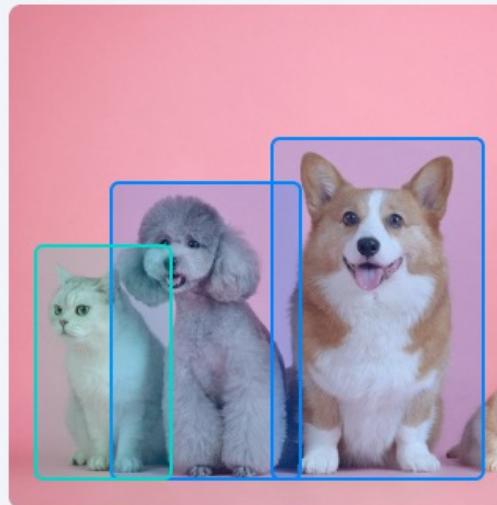


Cat



Single Object

Detection



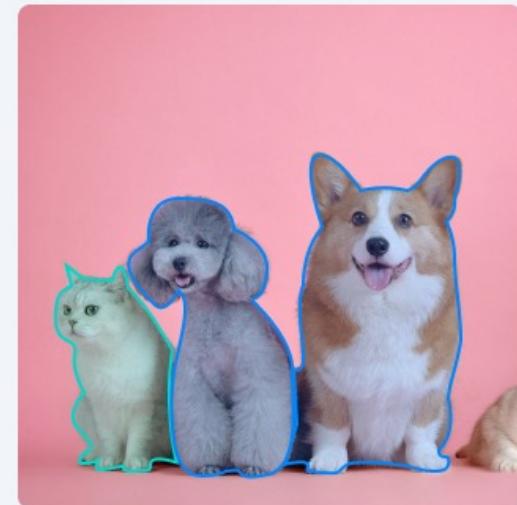
Cat

Dog



Multiple Objects

Segmentation



Cat

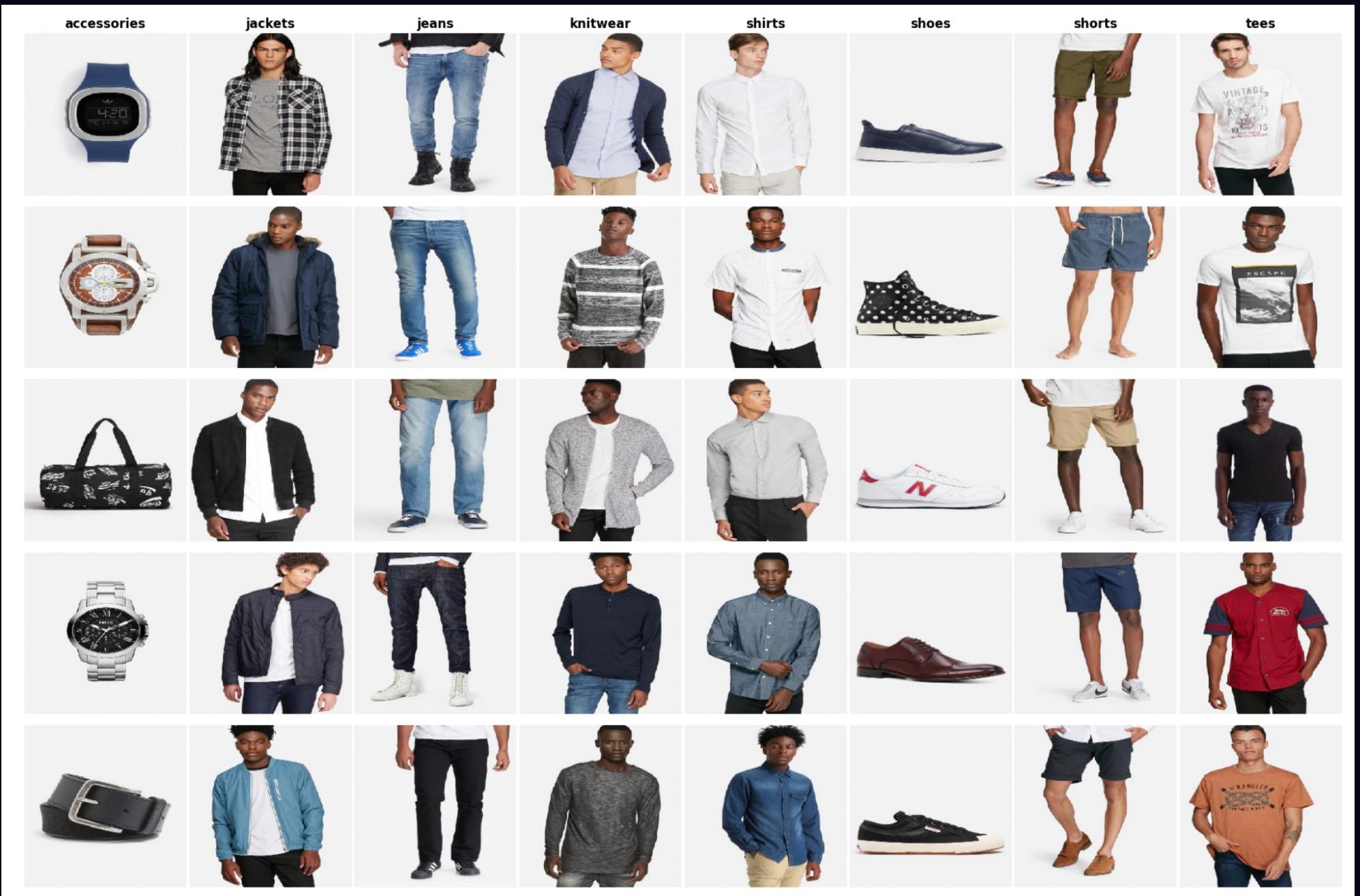
Dog



Jupyter Notebook

Classification and Visual Search.ipynb





```
Epoch 1/10
109/109 0s 641ms/step - accuracy: 0.7584 - loss: 0.7600
Epoch 1: val_accuracy improved from -inf to 0.95026, saving model to best_model.keras
109/109 80s 717ms/step - accuracy: 0.7595 - loss: 0.7567 - val_accuracy: 0.9503 - val_loss: 0.1754
Epoch 2/10
109/109 0s 637ms/step - accuracy: 0.9577 - loss: 0.1332
Epoch 2: val_accuracy did not improve from 0.95026
109/109 78s 712ms/step - accuracy: 0.9576 - loss: 0.1332 - val_accuracy: 0.9503 - val_loss: 0.1359
Epoch 3/10
109/109 0s 721ms/step - accuracy: 0.9698 - loss: 0.0950
Epoch 3: val_accuracy improved from 0.95026 to 0.95288, saving model to best_model.keras
109/109 88s 810ms/step - accuracy: 0.9698 - loss: 0.0950 - val_accuracy: 0.9529 - val_loss: 0.1358
Epoch 4/10
109/109 0s 1s/step - accuracy: 0.9770 - loss: 0.0754
Epoch 4: val_accuracy did not improve from 0.95288
109/109 126s 1s/step - accuracy: 0.9770 - loss: 0.0754 - val_accuracy: 0.9450 - val_loss: 0.1363
Epoch 5/10
109/109 0s 753ms/step - accuracy: 0.9839 - loss: 0.0519
Epoch 5: val_accuracy did not improve from 0.95288
109/109 91s 833ms/step - accuracy: 0.9839 - loss: 0.0520 - val_accuracy: 0.9503 - val_loss: 0.1437
Epoch 6/10
109/109 0s 675ms/step - accuracy: 0.9854 - loss: 0.0472
Epoch 6: val_accuracy did not improve from 0.95288
109/109 81s 745ms/step - accuracy: 0.9854 - loss: 0.0472 - val_accuracy: 0.9450 - val_loss: 0.1460
Epoch 7/10
109/109 0s 663ms/step - accuracy: 0.9893 - loss: 0.0400
Epoch 7: val_accuracy improved from 0.95288 to 0.96073, saving model to best_model.keras
109/109 81s 739ms/step - accuracy: 0.9893 - loss: 0.0400 - val_accuracy: 0.9607 - val_loss: 0.1406
Epoch 8/10
109/109 0s 654ms/step - accuracy: 0.9887 - loss: 0.0438
Epoch 8: val_accuracy did not improve from 0.96073
109/109 79s 728ms/step - accuracy: 0.9887 - loss: 0.0438 - val_accuracy: 0.9555 - val_loss: 0.1354
Epoch 9/10
109/109 0s 674ms/step - accuracy: 0.9939 - loss: 0.0263
Epoch 9: val_accuracy did not improve from 0.96073
109/109 82s 753ms/step - accuracy: 0.9940 - loss: 0.0263 - val_accuracy: 0.9503 - val_loss: 0.1398
Epoch 10/10
109/109 0s 659ms/step - accuracy: 0.9965 - loss: 0.0214
Epoch 10: val_accuracy did not improve from 0.96073
109/109 80s 734ms/step - accuracy: 0.9965 - loss: 0.0214 - val_accuracy: 0.9503 - val_loss: 0.1457
```

```
: # Evaluate on validation set
loss, accuracy = model.evaluate(valid_dataset)
print(f'Test Accuracy: {accuracy*100:.2f}%')
```

```
12/12 8s 650ms/step - accuracy: 0.9492 - loss: 0.1535
Test Accuracy: 96.07%
```



Embeddings

TODO



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

Object Detection.ipynb



Original Image



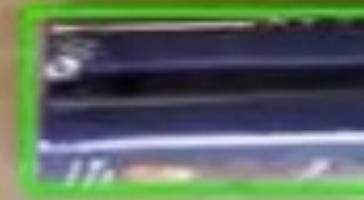
Detected 14 persons
Detected 4 cars

YOLOv8 Object Detection



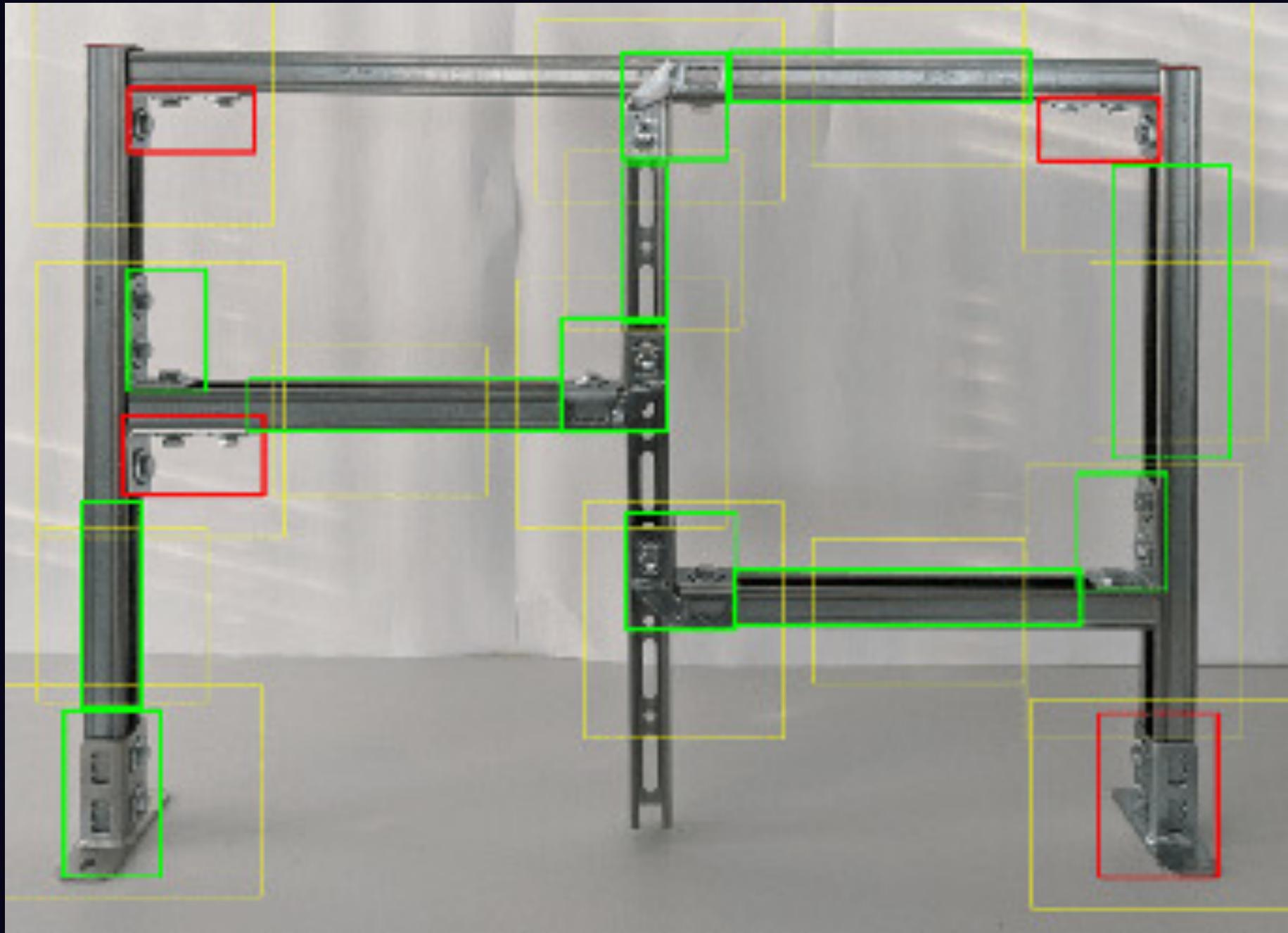
13:56:12

1



O





Jupyter Notebook

Segmentation.ipynb



Original Image



6541 pixels of type car
14417 pixels of type person

Segmentation Overlay



Original Image



Original Image with 50% Transparent Segmentation Mask



Car with Drop Shadow on New Background





Qwen2-VL-Max

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

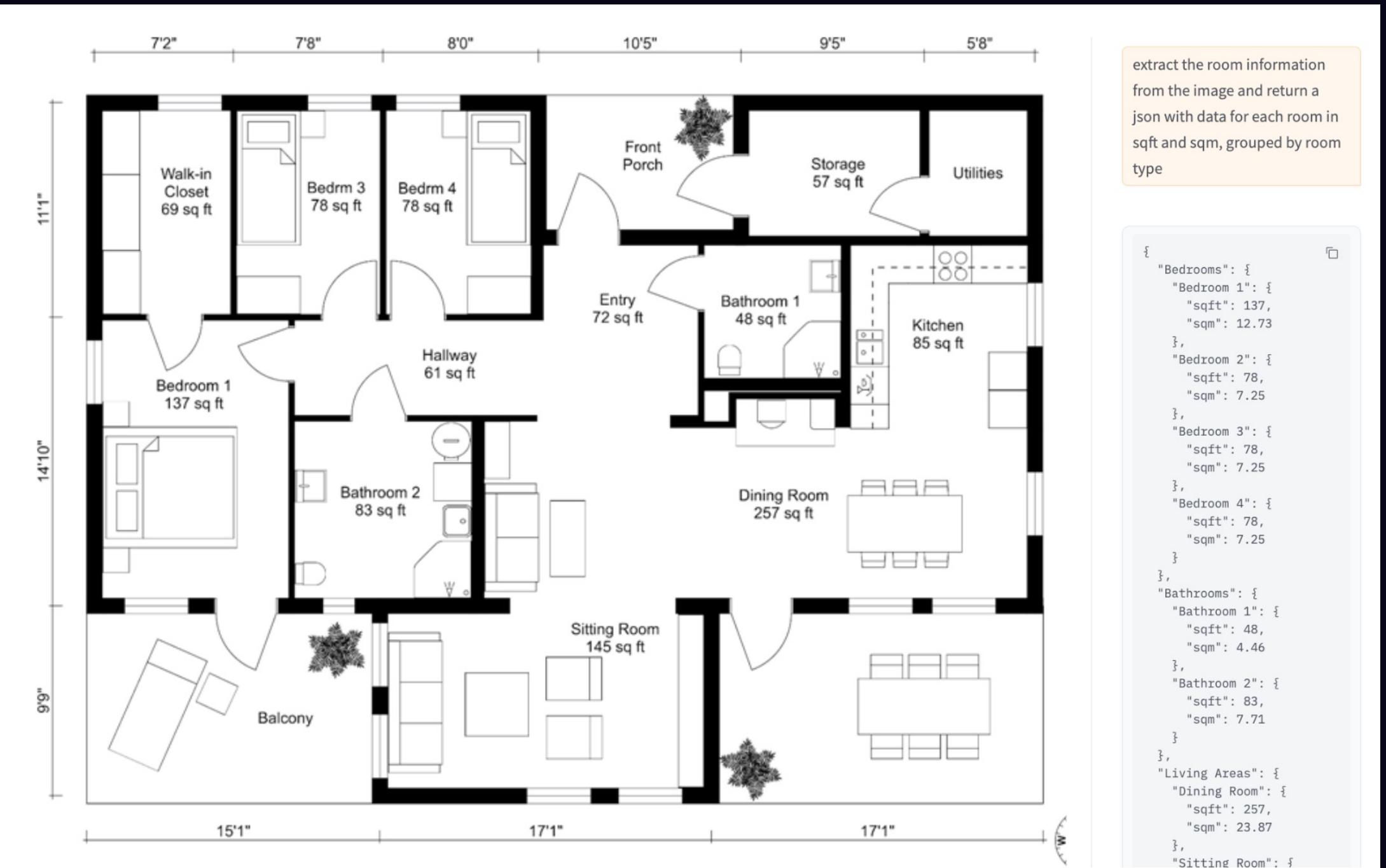
本WebUI基于Qwen2-VL-Max。

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"
  }
}
```





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"  
    ]  
  }  
}
```

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.

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-libarib24 --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

Jupyter Notebook

[https://github.com/mbamvianney/SPEECH-
EMOTION-RECOGNITION-WITH-MACHINE-
LEARNING](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



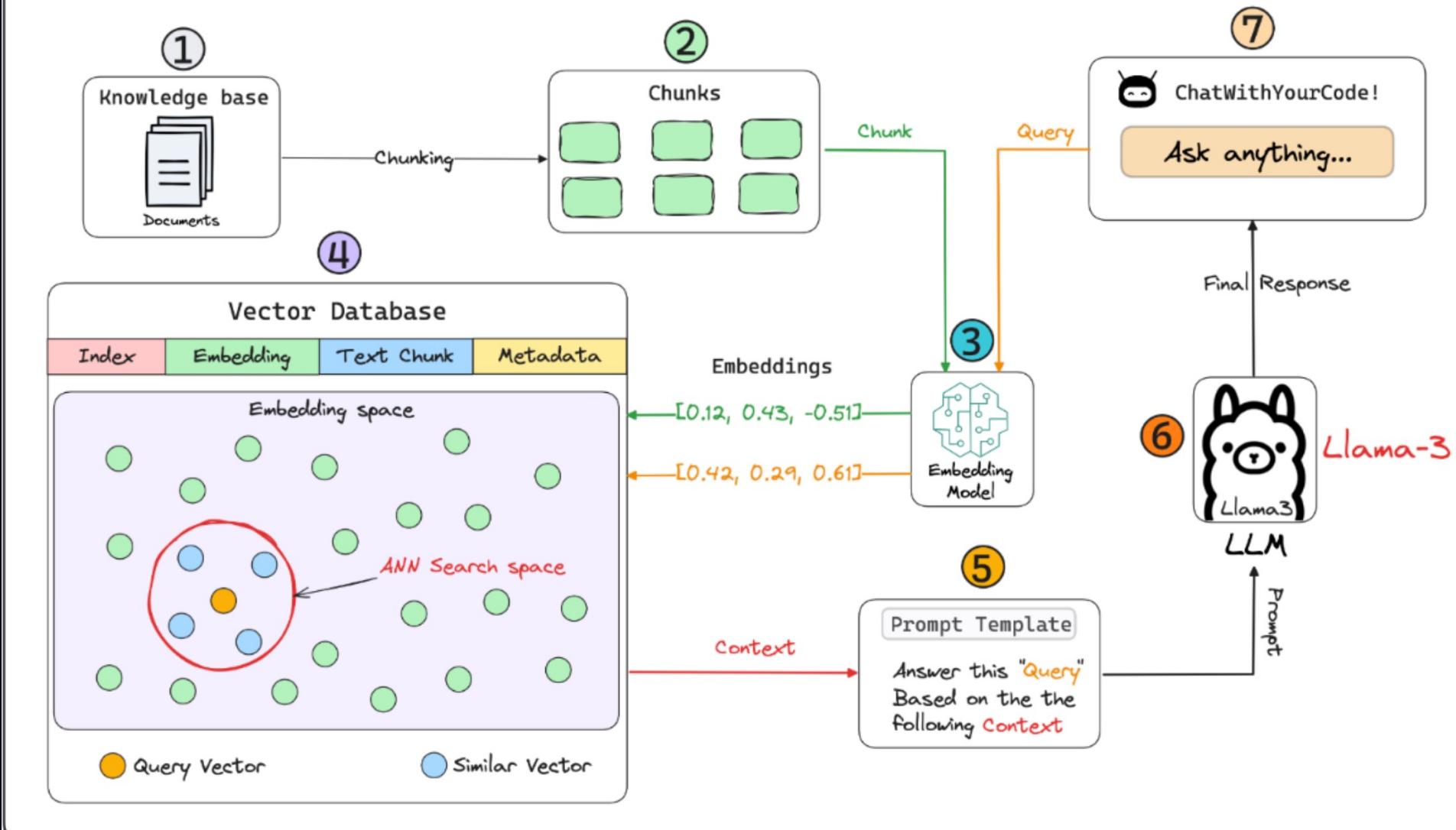
Voice Activity Detection



Text Data



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	



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/>

--- 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 | ♀ B |
~~ycombinator~~
W22 | won awards for AI projects from companies like Mercedes-Benz, NTT Japan, etc
location: Cape Town
blog: www.numberboost.com
email:

Use RAG

```
[7]: print(qa_chain.run("What is the main topic of the documents [<10 words]"))
```

The main topic of the documents is the abstracts and speaker details for several talks at a Python conference.

```
• [5]: print(qa_chain.run("Are there any talks about artificial intelligence?"))
```

Yes, there are several talks about artificial intelligence:

1. "Applying AI with Python" by Alex Conway. In this talk, the speaker will explore how Python's rich ecosystem of AI libraries and frameworks can be harnessed to quickly solve real-world problems. The session will demonstrate practical AI applications in real-world use cases that leverage Python's versatility. More details can be found [here](<https://2024.za.pycon.org/talks/11-applying-ai-with-python/>).

2. "Differentiation Engines: The Elves behind the AI Christmas" by Rajdeep Singh. This talk will provide a good overview of the different mathematical differentiation techniques and libraries within the Python ecosystem. It will also provide an in-depth look at what powers modern machine learning frameworks. More details can be found [here](<https://2024.za.pycon.org/talks/23-differentiation-engines-the-elves-behind-the-ai-christmas/>).

3. "Building a Decisioning Engine for Data Scientists: A Practical Guide" by Sholto Armstrong. This talk will introduce a new open-source decisioning framework built on Hamilton, a versatile framework designed to streamline the creation and management of dataflows using standard Python functions. More details can be found [here](<https://2024.za.pycon.org/talks/37-building-a-decisioning-engine-for-data-scientists-a-practical-guide/>).

```
• [6]: 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





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



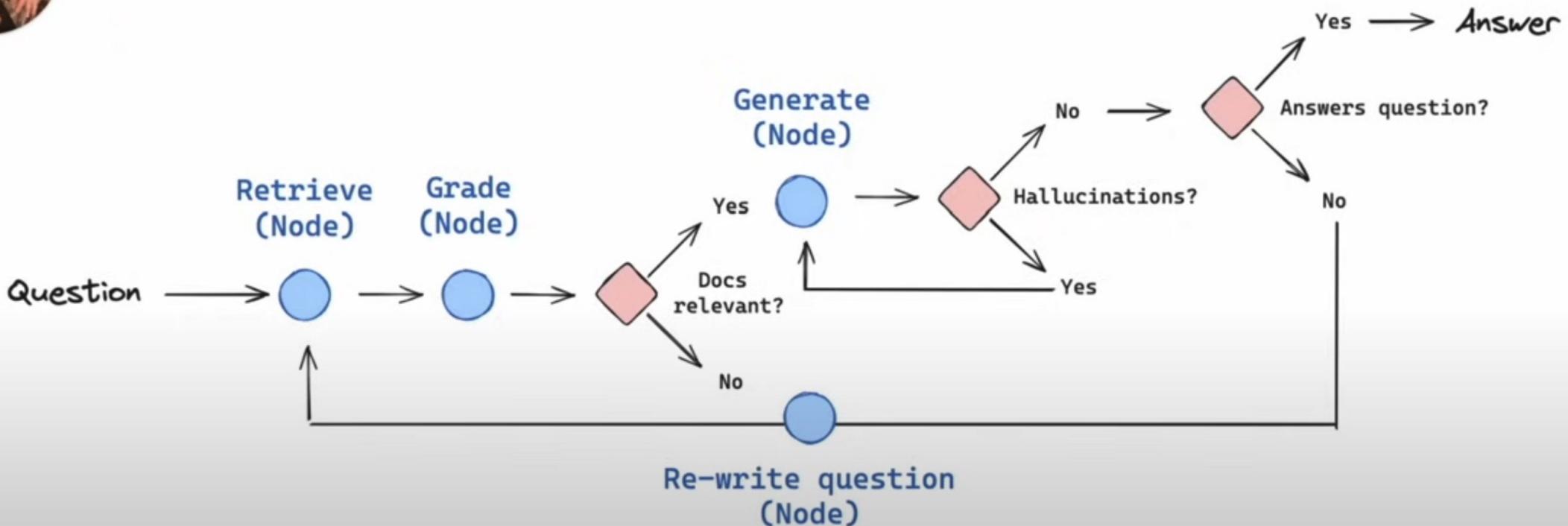
Sentence Similarity



AI Agents



Reasoning: Use reasoning / self-reflection around RAG



Our Agent Architecture

- Todo: diagram
- Agent
 - LLM checks if SQL or LLM should answer
 - If SQL
 - LLM writes SQL query with knowledge of database schema
 - Execute SQL
 - LLM takes SQL results and formulates markdown answer with analysis
 - If LLM
 - LLM writes answer using RAG
 - return answer



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)
```



DB Browser for SQLite - /Users/alexc/Documents/_meetups/pyconza 2024/code/Text/data/confere

New Database Open Database Write Changes Revert Changes Undo Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragmas Execute SQL

github github github

Table: **github**

Filter in any column

	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	2024-09-28T17:04:
2	20	https://github.com/gijzelaerr	gijzelaerr	Gjjs Molenaar	172	748	python-snap7	648	2024-09-17T16:20:
3	21	https://github.com/luisdza	luisdza	Luis de Sousa	8	3	aerialmzansi-website	2	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-pepele	adeline-pepele	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	nitheos	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	2	2024-09-26T11:53:
24	4	https://github.com/ruankie	ruankie	Ruan Pretorius	71	112	ecrival	68	2024-10-02T00:42:

Jupyter Notebook

Agent.ipynb



Imports

+ 4 cells hidden

Load Vector Store

+ 1 cell hidden

Setup RAG

+ 3 cells hidden

Setup SQL Database

+ 3 cells hidden




```
: def get_prompt_write_SQL_query_for_question(question):
    return f"""
        You are a Senior SQL Programmer who can understand the complex table structures and form SQL queries as per the user's question.

        Your job is to write a SQL query that answers the question and return the SQL code for the query and nothing else.

        The QUESTION is:
        {question}

        IMPORTANT instructions:
        1) Dont use JOIN if it is not necessary, check the table and its usable columns, if using JOIN do it with extra CAUTION
        2) Dont forget to apply inverted commas when required.
        3) Give no explanation and directly provide answers.
        4) Take care when joining tables, use right set of table-column configuration
        5) Return only a SQL query and nothing else

        Here is the table structure of the sql data:

        table_name      column_name data_type
0   github          speaker_id   INTEGER
1   github          github_url    TEXT
2   github          username     TEXT
3   github          name        TEXT
4   github          | public_repos INTEGER
5   github          total_stars  INTEGER
6   github          most_starred_repo_name TEXT
7   github          most_starred_repo_stars INTEGER
8   github          last_commit_datetime TEXT
9   github          hours_since_last_commit REAL
10  github          followers    INTEGER
11  github          following   INTEGER
12  github          created_at   TEXT
13  github          company     TEXT
14  github          bio         TEXT
15  github          location    TEXT
16  github          blog        TEXT
17  github          email       TEXT
18  speakers         speaker_id   INTEGER
19  speakers         name        TEXT
20  speakers         profile_url TEXT
21  speakers         photo_url   TEXT
22  speakers         twitter     TEXT
23  speakers         github      TEXT
24  speakers         fediverse   TEXT
25  speakers         bio_part_1  TEXT
26  speakers         bio_part_2  TEXT
27  speakers         bio_part_3  TEXT
28  speakers         bio_part_4  INTEGER
```



```
def get_prompt_analyze_sql(question, sql_results):
    return f'''
You are an expert analyst who can answer questions using supporting data which was returned from a SQL query on a database

Your job is to answer the input QUESTION, using the SQL_RESULTS in your answer and identifying any insights or trends yo

The QUESTION is:
{question}

The SQL_RESULTS are:
{sql_results}

Important instructions
* do not reference the SQL database or use the term SQL unless it's from a part of the data itself
* be concise and try to make points with short bullet points
* return the sql results dataset as part of your response

'''
```



```
: def agent(question = "How many talks in each category?", debug=False):
    # 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)

    if question_type == "SQL":
        # 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)
        if debug:
            print("SQL query type")
            print(sql_query)

        # scrub sqlquery
        sql_query=sql_query.replace("```sql\n","",).replace("\n``", "").replace("\n", " ")

        # query sql database with generated 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)

        return print_markdown(run_chat_query(prompt_analyze_sql))
    else:
        if debug:
            print("LLM query type")
        prompt_analyze_llm = get_prompt_analyze_llm(question)
        return print_markdown(run_chat_query(prompt_analyze_llm))
```



Conclusion



How can I apply AI with python?

To apply AI with Python, you can focus on several key areas, each leveraging specific libraries and techniques. Here's a concise breakdown with examples:

1. Machine Learning

- **Libraries:** scikit-learn, TensorFlow, Keras, PyTorch
- **Example:** Building a classifier to predict outcomes based on input data.

```
python
Copy code

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import load_iris

# Load dataset
iris = load_iris()
X_train, X_test, y_train, y_test = train_test_split(iris.data, iris.target, test_size=0.2, random_state=42)

# Train model
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Evaluate model
accuracy = model.score(X_test, y_test)
print(f"Accuracy: {accuracy}")
```

2. Natural Language Processing (NLP)

- **Libraries:** spaCy, Transformers, nltk, GPT, BERT
- **Example:** Text classification or sentiment analysis using a pretrained transformer model.

2. Natural Language Processing (NLP)

- **Libraries:** spaCy, Transformers, nltk, GPT, BERT
- **Example:** Text classification or sentiment analysis using a pretrained transformer model.

python

Copy code

```
from transformers import pipeline

# Load sentiment-analysis model
classifier = pipeline('sentiment-analysis')

# Analyze text
result = classifier("I love AI with Python!")
print(result)
```

3. Computer Vision

- **Libraries:** OpenCV, TensorFlow, PyTorch, YOLO
- **Example:** Detecting objects in images or videos.

python

Copy code

```
import cv2

# Load image
image = cv2.imread('image.jpg')

# Convert to grayscale
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Apply Canny edge detection
edges = cv2.Canny(gray, 100, 200)

# Show result
```

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



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



Thank You!

Alex Conway

alex@numberboost.com

x.com/alxchnwy