

[Open in app](#)

Search



★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



WANT TO LEARN AI?

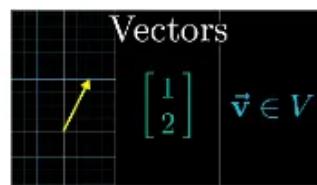
# Roadmap to Learn AI in 2024

A free curriculum for hackers and programmers to learn AI

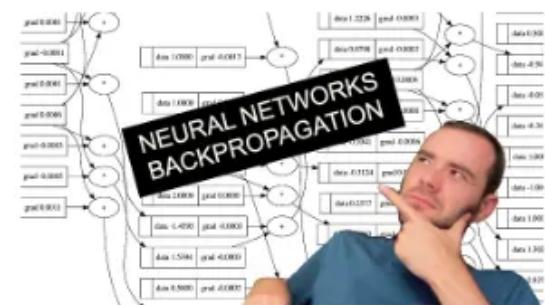
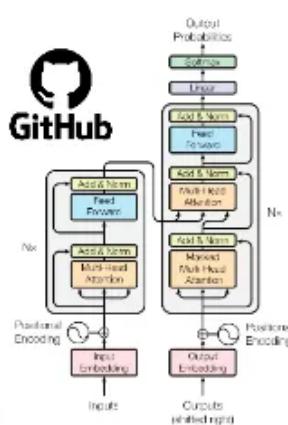
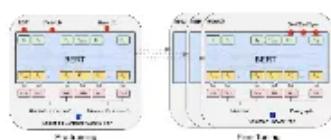
Benedict Neo · [Follow](#)

Published in bitgrit Data Science Publication

11 min read · Feb 9, 2024

[Listen](#)[Share](#)[More](#)

# fast.ai

[kaggle](#)[arXiv](#)[DeepLearning.AI](#)

Gemini



Llama 2


**MISTRAL**  
**AI\_**

7



made with excalidraw

If you found this roadmap helpful, follow me on [Twitter](#) and [LinkedIn](#)! Also subscribe to this [newsletter](#) where I share interesting links every week with my friend.

**S**o, you want to learn AI? But you don't know how or where to get started?

I wrote the [Top 20 free Data Science, ML, and AI MOOCs on the Internet](#) back in 2020. But I've realized that doing many courses isn't the way.

To escape tutorial hell and **really learn**, you have to get hands-on, write algorithms from scratch, implement papers, and do fun side projects using AI to solve problems.



This article attempts to craft a **free** curriculum that follows that philosophy. I'm working on some of these courses, so [reach out](#) on [Twitter](#) or [LinkedIn](#) if you want to learn together!

Also, leave a comment if you think it's missing anything!

But first, a few notes on the curriculum and some advice on learning.

## Top-down approach

This curriculum follows a top-down approach — **code first, theory later**.

I like to **learn out of necessity**. So, if I have to figure out something, a problem to solve, or a prototype to make, I will reach far and wide for the information I need, study, make sense of it, and then act on it.

For example, I aim to be an [AI engineer](#) who understands LLMs at a fundamental level, which involves having the skill to code transformers from scratch and fine-tuning LLMs on GPUs, etc. I can't do that now because there are gaps in my knowledge, and I aim to fill in those gaps.

It is also NLP-focused; if you're looking for other AI specializations like computer vision or reinforcement learning, comment below or DM me on [Twitter](#) or [LinkedIn](#). I will pass you some recommendations.

Before I dump a bunch of links on you, I wish somebody had told me **two important things** before I started learning anything.

## Learn in Public

There's a lot to learn, and you will never be done learning, especially with AI, when new revolutionary papers and ideas are released weekly.

The biggest mistake you can make is to learn in private. You don't create any opportunities for yourself if you do that. You don't have anything to show for it besides being able to say you completed something. What matters more is what you made of the information, how you turned it into knowledge to be shared with the public, and what novel ideas and solutions came from that information.

So, you should learn in public.



That means having a **habit of creating**.

This can mean:

- writing blogs and tutorials
- join hackathons and collaborate with others
- ask and answer questions in Discord communities
- work on side projects you're passionate about
- tweeting about something interesting you discovered new

And speaking about Twitter,

## Use Twitter

If you follow the right people and use it right, Twitter is the highest-value social platform anyone can be on today.

Who to follow? See this AI list by Suhail.

How to use Twitter? Read Near's How to Twitter Successfully.

DM people on Twitter. Be sincere, keep it short, and have a specific ask. This guide on How to write a cold email by Sriram Krishnan can also apply to DMs.

**How to tweet? Read [Anatomy of a Tweet](#) by Jason, creator of [Instructor](#), who grew from 0 → 14k followers in months.**

If you're reading this, [follow me on Twitter!](#)

[DM me](#) about what you're up to! I'm always up for collaborating on cool projects.

Now let's get into it.

## Table of contents

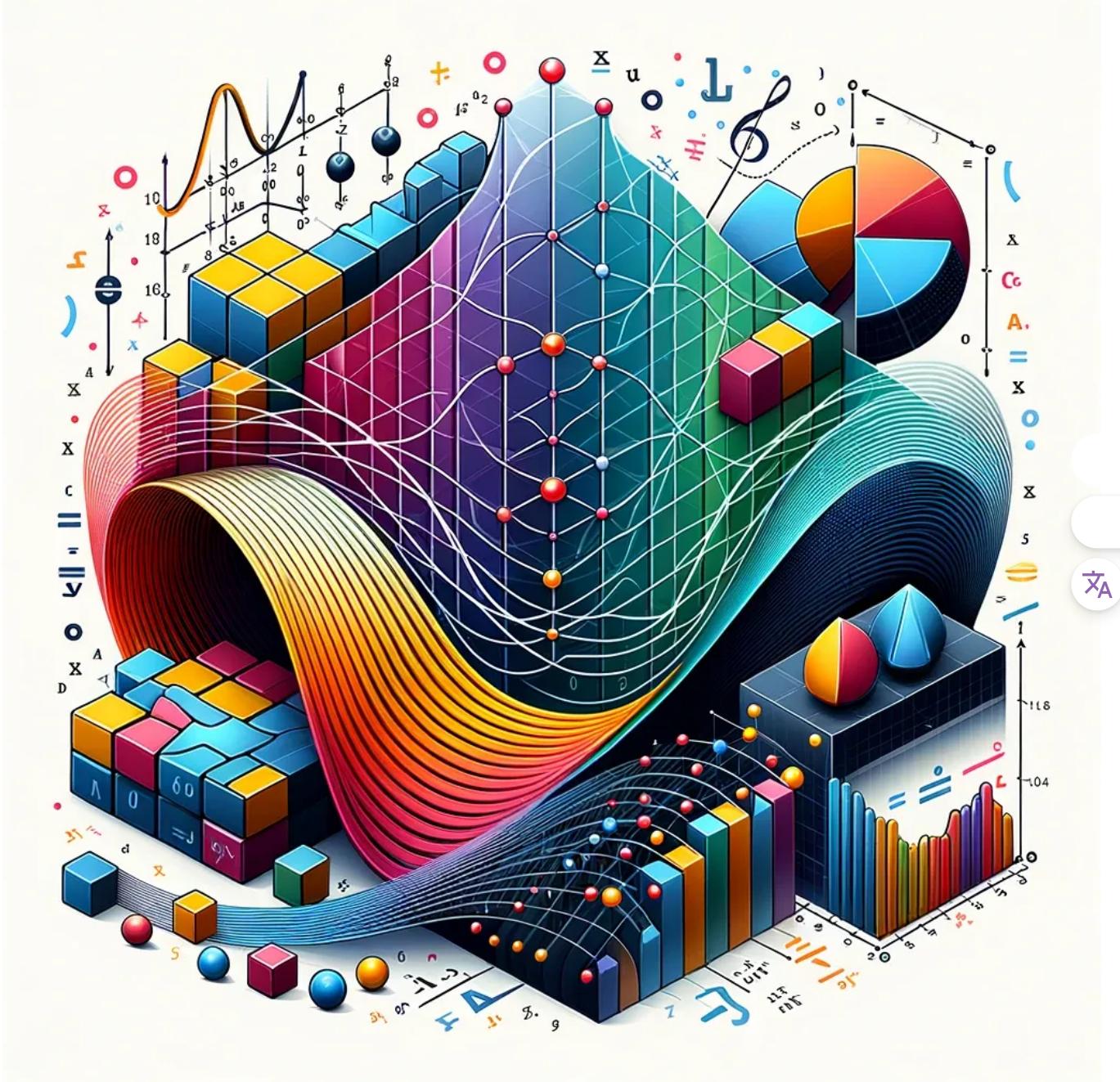


- [Mathematics](#)
- [Tools](#)
  - [Python](#)
  - [PyTorch](#)
- [Machine Learning](#)
  - [Write from Scratch](#)
  - [Compete](#)
  - [Do side projects](#)
  - [Deploy them](#)
  - [Supplementary](#)
- [Deep Learning](#)
  - [Fast.ai](#)
  - [Do more competitions](#)
  - [Implement papers](#)
  - [Computer Vision](#)
  - [NLP](#)
- [Large Language Models](#)
  - [Watch Neural Networks: Zero to Hero](#)
  - [Free LLM boot camp](#)
  - [Build with LLMs](#)
  - [Participate in hackathons](#)
  - [Read papers](#)
  - [Write Transformers from scratch.](#)

- Some good blogs
- Watch Umar Jamil
- Learn how to run open-source models.
- Prompt Engineering
- Fine-tuning LLMs
- RAG
- How to stay updated
- Other curriculums/listicles you may find useful

## Mathematics





DALL-E

Machine learning relies heavily on three pillars of mathematics: linear algebra, calculus, probability, and statistics. Each plays a unique role in enabling algorithms to function effectively.

- **Linear Algebra:** the mathematical toolkit for data representation and manipulation, where matrices and vectors form the language for algorithms to interpret and process information
- **Calculus:** The engine for optimization in machine learning, enabling algorithms to learn and improve by understanding gradients and rates of change.

- **Probability and Statistics:** The foundation for decision-making under uncertainty, allowing algorithms to predict outcomes and learn from data through models of randomness and variability.

This is a great series on Math for ML from a programmer's perspective: [Math for Machine Learning by Weights & Biases \(code\)](#)

If you want a code-first approach to Linear Algebra, do [Computational Linear Algebra \(video, code\)](#) by the creators of fast.ai.

Read [Introduction to Linear Algebra for Applied Machine Learning with Python](#) alongside the course.

If you want something more traditional, look at Imperial College London lectures ·  [Linear Algebra & Multivariate Calculus](#).

Watch 3Blue1Brown's [Essence of Linear Algebra](#) and [Essence of Calculus](#).

Watch [Statistics Fundamentals](#) by StatQuest for statistics

## Supplementary

- Book: [Mathematics for Machine Learning](#)
- Paper: [The Matrix Calculus You Need For Deep Learning](#)

## Tools



DALL-E

## Python

Beginners start here: [Practical Python Programming](#).

If you're already comfortable with Python, do this [Advanced Python Mastery](#).

They're both great courses by David Beazley, author of [Python Cookbook](#).

After that, watch some of [James Powell's talks](#)

Read [Python Design Patterns](#).

## Supplementary

- Book: [Fluent Python, 2nd Edition \(code\)](#)
- Podcasts: [Real Python & Talk Python](#)

## PyTorch

Watch [PyTorch Tutorials](#) by [Aladdin Persson](#)

The PyTorch website is a great place to be.

- [PyTorch Examples](#)
- [Official PyTorch Tutorials](#)
- [FAQ page](#)

Test your knowledge with some puzzles

- [srush/Tensor-Puzzles: Solve puzzles. Improve your PyTorch](#)

## Supplementary

- Book: [Programming PyTorch for Deep Learning](#)



## Machine Learning



DALL·E

Read the [100-page ML book](#).

### Write from Scratch

While you're reading, write the algorithms from scratch.

Look at the repositories below

- [eriklindernoren/ML-From-Scratch](#)
- [JeremyNixon/oracle](#)
- [trekhleb/homemade-machine-learning](#)

If you want a challenge, write PyTorch from scratch by following this course.

- [MiniTorch: A DIY Course on Machine Learning Engineering \(videos, code\)](#)

## Compete

Apply what you learn in competitions.

- Join ML competitions on platforms like [bitgrit](#) and [Kaggle](#); find more in [this article](#).
- Look at [past winning solutions](#) and study them



## Do side projects

Read [Getting machine learning to production](#) by Vicki Boykis

She also wrote about what she learned building [Viberary](#), a semantic search for books.

Get a dataset and build a model (i.e., use [earthaccess](#) to get NASA Earth data).

Create a UI with [streamlit](#) and share it on Twitter.

## Deploy them

Get the models in production. Track your experiments. Learn how to monitor models. Experience data and model drift firsthand.

Here are some excellent resources

- [Made With ML](#)
- [DataTalksClub/mlops-zoomcamp: Free MLOps course](#)
- [chiphuyen/machine-learning-systems-design](#)
- [Evidently AI — ML system design: 300 case studies](#)
- [stas00/ml-engineering: Machine Learning Engineering Online Book](#)

## Supplementary

- [Machine Learning with PyTorch and Scikit-Learn \(code\)](#)
- [\[1811.12808\] Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning](#)
- [Introduction to Machine Learning Interviews Book · MLIB](#)

## Deep Learning



If you want top-down, start with fast.ai.

### Fast.ai

- fast.ai ([part1](#), [part2](#)) + [W&B Study Group](#)

Liked fast.ai? Check out [Full Stack Deep Learning](#).

If you want a more comprehensive, traditional course, check out [UNIGE 14x050 — Deep Learning by François Fleuret](#).

If you need to reach for theory at some point, these are great books.

- [Dive into Deep Learning](#) (has code examples in PyTorch, NumPy/MXNet, JAX, and TensorFlow)
- [Deep Learning](#) by Ian Goodfellow, Yoshua Bengio, and Aaron Courville
- [Neural networks and deep learning](#)
- [Understanding Deep Learning](#) (with hands-on [notebooks](#))

Read [The Little Book of Deep Learning](#) on your phone instead of scrolling Twitter.

Read these while your neural networks are converging.

- [A Recipe for Training Neural Networks](#)
- [Deep Neural Nets: 33 years ago and 33 years from now](#)



## Do more competitions

- [PlantTraits2024 — FGVC11 | Kaggle](#) (computer vision)

## Implement papers

Check out [labml.ai Annotated PyTorch Paper Implementations](#)

Papers with Code is a great resource; here's [BERT explained](#) on their website.

Below are some resources for the specializations within Deep Learning

### Computer Vision

A lot of people recommend [CS231n: Deep Learning for Computer Vision](#). It's challenging but worth it if you get through it.

### Reinforcement Learning

For RL, these two are great:

- [Spinning Up in Deep RL](#) by OpenAI
- 😊 [Deep Reinforcement Learning Course — Hugging Face](#)

### NLP

Another great Stanford course, [CS 224N | Natural Language Processing with Deep Learning](#)

Learn Hugging Face: [Hugging Face NLP Course](#)

Check out this [Super Duper NLP Repo](#)

## Good articles and breakdowns

- [BERT Research – Ep. 1 – Key Concepts & Sources · Chris McCormick](#)
- [The Illustrated Word2vec – Jay Alammar](#)
- [The Illustrated BERT, ELMo, and co. \(How NLP Cracked Transfer Learning](#)
- [Understanding LSTM Networks – colah's blog](#)
- [PyTorch RNN from Scratch – Jake Tae](#)

## Supplementary

- [Natural Language Processing with Transformers Book](#)



## Large Language Models



First, watch [\[1hr Talk\] Intro to Large Language Models](#) by Andrej.

Then [Large Language Models in Five Formulas](#), by [Alexander Rush — Cornell Tech](#)

**Watch Neural Networks: Zero to Hero**

It starts with explaining and coding backpropagation from scratch and ends with writing GPT from scratch.

## Neural Networks: Zero To Hero by Andrej Karpathy

He just released a new video → Let's build the GPT Tokenizer

You can also look at GPT in 60 Lines of NumPy | Jay Mody while you're at it.

## **Free LLM boot camp**

A paid LLM Bootcamp released for free by Full Stack Deep Learning.

It teaches prompt engineering, LLMOps, UX for LLMs, and how to launch an LLM app in an hour.

Now that you're itching to build after this boot camp,



## **Build with LLMs**

Want to build apps with LLMs?

Watch Application Development using Large Language Models by Andrew Ng

Read Building LLM applications for production by Huyen Chip

As well as Patterns for Building LLM-based Systems & Products by Eugene Yan

Refer to the OpenAI Cookbook for recipes.

Use Vercel AI templates to get started.

## **Participate in hackathons**

lablab.ai has new AI hackathons every week. Let me know if you want to team up!

If you want to go deeper into the theory and understand how everything works:

## **Read papers**

A great article by Sebastian Raschka on Understanding Large Language Models, where he lists some papers you should read.

He also recently published another article with papers you should read in January 2024, covering mistral models.

Follow his substack [Ahead of AI](#).

## Write Transformers from scratch.

Read [The Transformer Family Version 2.0 | Lil'Log](#) for an overview.

Choose whichever format suits you best and implement it from scratch.

### Paper

- [Attention Is All You Need](#)
- [The Illustrated Transformer](#)
- [The Annotated Transformer](#) by Harvard
- [Thinking like Transformer](#)



### Blogs

- [Creating a Transformer From Scratch – Part One: The Attention Mechanism \(part 2\) \(code\)](#)
- [Understanding and Coding the Self-Attention Mechanism of Large Language Models From Scratch](#) by [Sebastian Raschka, PhD](#)
- [Transformers from scratch](#)

### Videos

- [Coding a Transformer from scratch on PyTorch, with full explanation, training and inference](#)
- [NLP: Implementing BERT and Transformers from Scratch](#)

You can code transformers from scratch now. But there's still more.

Watch these [Stanford CS25 – Transformers United](#) videos.

### Some good blogs

- [Gradient Descent into Madness – Building an LLM from scratch](#)
- [The Illustrated Transformer – Jay Alammar](#)
- [Some Intuition on Attention and the Transformer](#) by Eugene Yan

- [Speeding up the GPT – KV cache | Becoming The Unbeatable](#)
- [Beyond Self-Attention: How a Small Language Model Predicts the Next Token](#)
- [Llama from scratch \(or how to implement a paper without crying\) | Brian Kitano](#)
- [Improving LoRA: Implementing Weight-Decomposed Low-Rank Adaptation \(DoRA\) from Scratch](#)

### **Watch Umar Jamil**

He has fantastic in-depth videos explaining papers. He also shows you the code.

- [LoRA: Low-Rank Adaptation of Large Language Models – Explained visually + PyTorch code from scratch](#)
- [Mistral / Mixtral Explained: Sliding Window Attention, Sparse Mixture of Experts, Rolling Buffer](#)
- [Attention is all you need \(Transformer\) – Model explanation \(including math\), Inference and Training](#)
- [LLaMA explained: KV-Cache, Rotary Positional Embedding, RMS Norm, Grouped Query Attention, SwiGLU](#)
- [Retrieval Augmented Generation \(RAG\) Explained: Embedding, Sentence BERT, Vector Database \(HNSW\)](#)



Some more links related to LLMs that are not exhaustive. Look at [LLM Syllabus](#) for a more comprehensive syllabus for LLMs.

### **Learn how to run open-source models.**

Use [ollama: Get up and running with Llama 2, Mistral, and other large language models locally](#)

They recently released [Python & JavaScript Libraries](#)

### **Prompt Engineering**

Read [Prompt Engineering | Lil'Log](#)

[ChatGPT Prompt Engineering for Developers](#) by Ise Fulford (OpenAI) and Andrew Ng

DeepLearning.ai also has other [short](#) courses you can enroll in for free.

## Fine-tuning LLMs

Read the [Hugging Face fine-tuning guide](#).

A good guidebook: [Fine-Tuning — The GenAI Guidebook](#)

Check out [axolotl](#).

This is a good article: [Fine-tune a Mistral-7b model with Direct Preference Optimization](#) | by Maxime Labonne

## RAG

A great article by Anyscale: [Building RAG-based LLM Applications for Production](#)

A comprehensive overview of [Retrieval Augmented Generation](#) by Aman Chadha



## How to stay updated

Combination of newsletters + podcasts + Twitter

For papers, you can follow [AK \(@\\_akhaliq\)](#)

For podcasts, the best I've found is [Latent Space](#) by Swyx & Alessio

Join their [Discord](#).

They also have this newsletter, [Smol Talk](#), which summarizes all big AI discords.

Some other newsletters I like are:

- [The Batch | DeepLearning.AI | AI News & Insights](#)
- [Deep Learning Weekly](#)
- [Interconnects | Nathan Lambert](#)
- [AI Tidbits | Sahar Mor](#)

More in [this article](#).

## Other curriculums/listicles you may find useful.

My list was not meant to be exhaustive, but if you still want to find more, here are some.

- [openai/syllabus.md](#)
- [AI Canon | Andreessen Horowitz](#)
- [AI Learning Curation – LLM Utils](#)
- [Threshold to the AI Multiverse | Open DeepLearning](#)
- [louisfb01/start-langs: A complete guide to start and improve your LLM skills in 2023](#)



I've spent enough time writing and organizing this that it's diminishing returns. It's time to learn and build.

I hope this will help you in your AI journey!

If you've read this far, don't forget to [reach out](#) or leave a comment :)

Be sure to follow the [bitgrit Data Science Publication](#) to keep updated!

Want to discuss the latest developments in Data Science and AI with other data scientists? [Join our discord server!](#)

Follow Bitgrit below to stay updated on workshops and upcoming competitions!

[Discord](#) | [Website](#) | [Twitter](#) | [LinkedIn](#) | [Instagram](#) | [Facebook](#) | [YouTube](#)

Artificial Intelligence

AI

Machine Learning

Deep Learning

Python

[Follow](#)

## Written by Benedict Neo

10.4K Followers · Editor for bitgrit Data Science Publication

Python, Data Science, and AI

## More from Benedict Neo and bitgrit Data Science Publication



 Benedict Neo in bitgrit Data Science Publication

### Building an Image Classification model with PyTorch from scratch

A step-by-step guide to building a CNN model with PyTorch.

18 min read · Apr 23, 2021

 553

 1

 +

...

```
def combinations(iterable, r):
    # combinations('ABCD', 2) --> AB AC AD BC BD CD
    # combinations(range(4), 3) --> 012 013 023 123
    pool = tuple(iterable)
    n = len(pool)
    if r > n:
        return
    indices = list(range(r))
    yield tuple(pool[i] for i in indices)
    while True:
        for i in reversed(range(r)):
            if indices[i] != i + n - r:
                break
        else:
            return
        indices[i] += 1
        for j in range(i+1, r):
            indices[j] = indices[j-1] + 1
```



Benedict Neo in bitgrit Data Science Publication



## 12 Python Features Every Data Scientist Should Know

Master the essential Python fundamentals

6 min read · May 24, 2023



1.3K



13



Benedict Neo in bitgrit Data Science Publication

## 4 Ways to Time Python Functions

## Which one should you use?

6 min read · Feb 23, 2023

👏 162

💬 1



...



+



Benedict Neo in Geek Culture

## 4 Free Tools to Summarize YouTube Videos with GPT

Learn faster and better from videos using these tools

4 min read · Apr 12, 2023

👏 459

💬 5

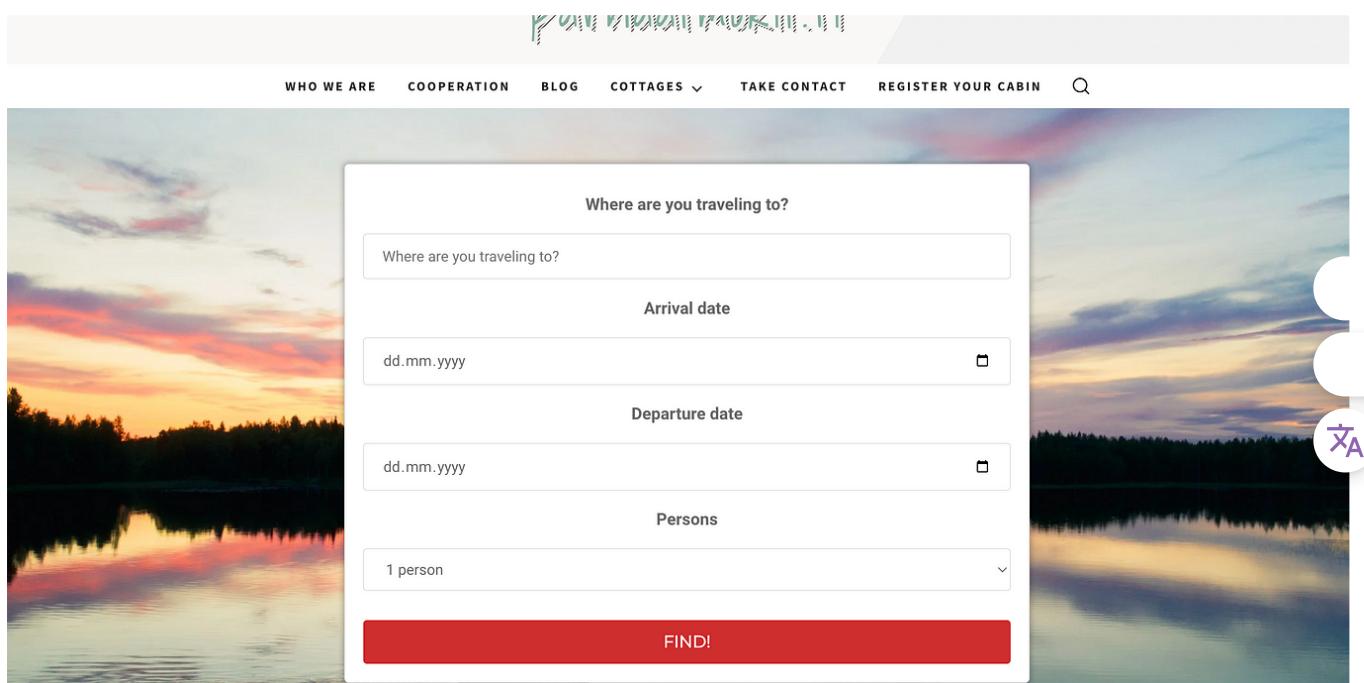


...

See all from Benedict Neo

See all from bitgrit Data Science Publication

## Recommended from Medium



The screenshot shows a travel search interface overlaid on a scenic background of a lake at sunset. The interface includes fields for destination, arrival date, departure date, and number of persons, along with a red "FIND!" button.

Where are you traveling to?

Arrival date

dd.mm.yyyy

Departure date

dd.mm.yyyy

Persons

1 person

FIND!

 Artturi Jalli

### I Built an App in 6 Hours that Makes \$1,500/Mo

Copy my strategy!

★ • 3 min read • Jan 23, 2024

 13.7K  160



...



Tirendaz AI in Level Up Coding



## How to Use ChatGPT in Daily Life?

Save time and money using ChatGPT

9 min read · Apr 4, 2023



4.7K

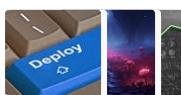


84



...

## Lists



### Predictive Modeling w/ Python

20 stories · 1001 saves



### Natural Language Processing

1282 stories · 773 saves



### Practical Guides to Machine Learning

10 stories · 1197 saves



### ChatGPT prompts

47 stories · 1264 saves



Mil Hoornaert in Long. Sweet. Valuable.



## Stop Listening to Music, It Will Change Your Life

I stopped listening to music, find out what the results and benefits are in this post!

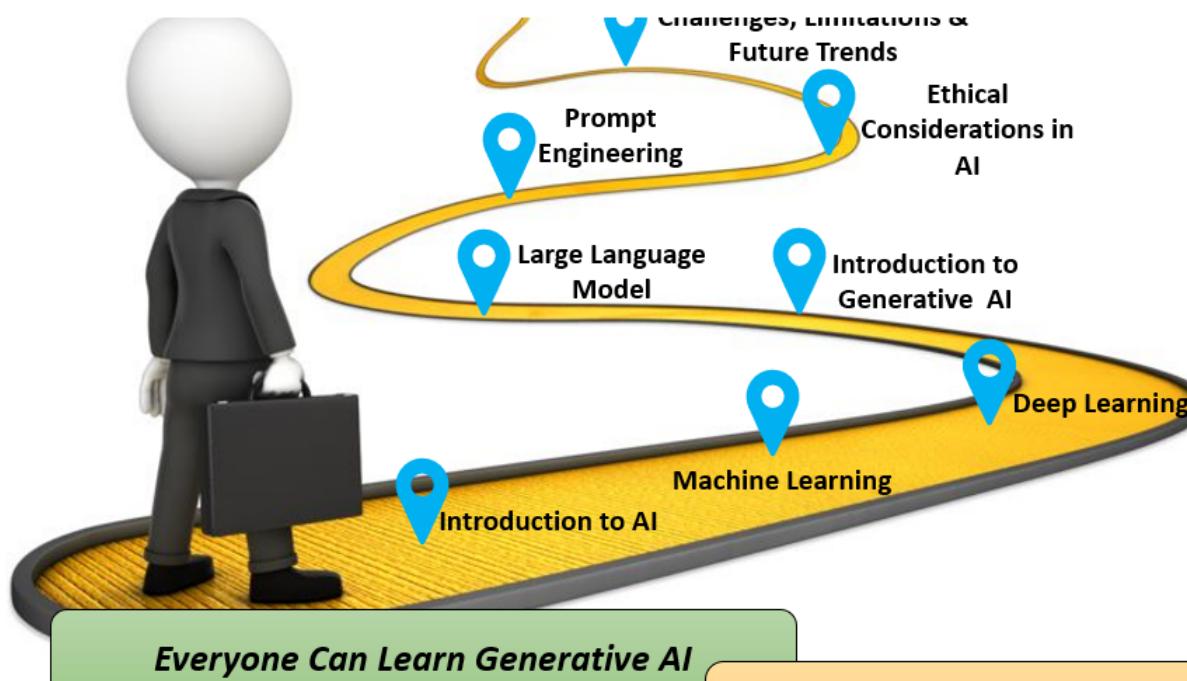
◆ · 5 min read · Nov 23, 2023

10.3K

401



...



Raja Gupta

## Generative AI for Beginners: Part 1—Introduction to AI

## Learn Generative AI by Spending 15 Minutes Daily for 8 Days

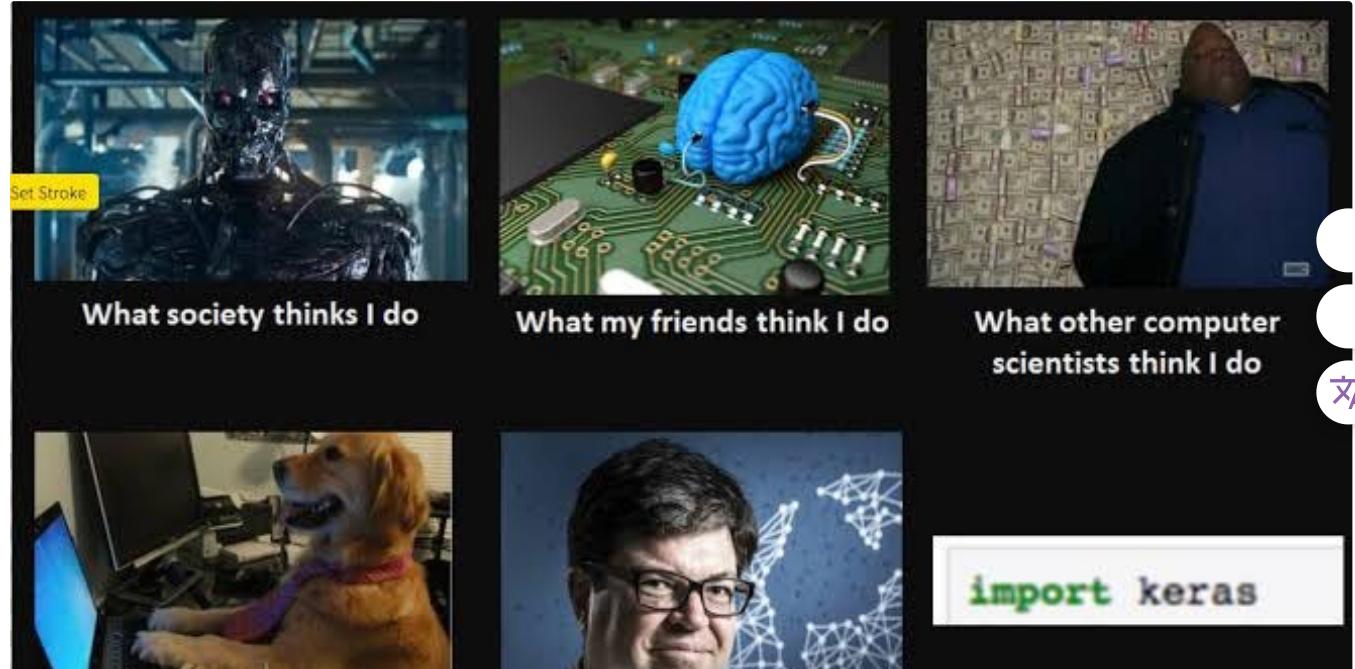
13 min read · Feb 8, 2024

👏 1.8K

💬 26



...



 Sibtain Wahab

## 50 Days to AI/ML: from Zero to Hero (for non-CS background)

Interested in AI? Non-CS background? Don't know where to start? Or what it takes to get there? Here you go!!!

4 min read · Feb 20, 2024

👏 487

💬 7



...



 Nathan Rosidi



## Data Science in 2024—What Has Changed

What has changed in the data science landscape, and what are the challenges of the 2024 data science job market?

4 min read · Jan 29, 2024

 939

 17



...

See more recommendations