

[Open in app](#)

Search

[Python in Plain Eng...](#) · [Follow publication](#)

This member-only story is on us. [Upgrade](#) to access all of Medium.

★ Member-only story

# Why Python Is No Longer My First Choice for Data Science

A personal journey through better tools, team dynamics, and letting go of old habits

5 min read · 1 day ago



PURRFECT SOFTWARE LIMITED

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



image generated by ideogram

There was a weird moment last year.

I was on a Zoom call with my team, presenting a machine learning pipeline I'd built in Python. I'd used all the usual suspects — `pandas`, `sklearn`, `xgboost`, `joblib` for saving models.

It worked... but just barely. Every change needed a careful re-run. Every refactor broke something. Every deploy felt like I was defusing a bomb.

After the demo, a junior teammate quietly asked, “**Could we maybe do this in dbt or Polars next time?**”

And I felt it. That twinge of being outdated. Like when you realize the new generation has better tools — and they’re not wrong.

I wasn’t offended. I was... relieved. Because I was tired too.

Python wasn't bad.

It just wasn't great anymore.

Not for everything.

## 9 Mind-Blowing Automation Scripts You Need To Try Using Python

Simple Python Projects That Make Life Easier, Smarter, and Way More Fun

medium.com

## The Team Shift: Not Everyone Wants to Code

One of the biggest lessons I've learned over the past few years is this:

*Not everyone on your team wants to — or should have to — write Python.*

I once worked on a team with:

- A data engineer who loved Scala
- An analyst who lived in SQL
- A PM who used Google Sheets like a database
- A machine learning engineer who barely touched Pandas but lived in TensorFlow

If I built everything in Python, I was the bottleneck.

When we moved to:

- dbt for data modeling

- Airflow for orchestration
- Streamlit for dashboards
- Snowflake for analysis

...everyone could contribute in their own way.

I started using Python only where it made sense: the weird stuff. The messy scripts. The logic-heavy parts. The edge cases. And it *worked better* that way.

I didn't have to convince anyone to "just learn Python." I could meet them where they were — and that made collaboration so much smoother.

### I Tried the Top 3 AI Video Generators of 2025 — Here's What Blew Me Away

A creator's deep dive into Sora, Gen-2, and Pika Labs, with real examples and honest impressions.

[blog.stackademic.com](http://blog.stackademic.com)

## Where Python Still Wins (At Least for Me)

Despite everything I've said, I'm not giving up Python. Not even close. There are places where Python *still rules* — and probably always will.

### 1. Rapid Prototyping

When I have a fuzzy idea — something like "what if we cluster users based on how they scroll through the app?" — I still go straight to Jupyter.

```
from sklearn.cluster import KMeans
import pandas as pd

df = pd.read_csv("user_behavior.csv")
model = KMeans(n_clusters=4).fit(df[["scroll_depth", "time_spent"]])
```

It's fast. It's messy. It works. That's Python at its best.

## **2. Machine Learning Libraries**

Let's be real: PyTorch, TensorFlow, Hugging Face — these ecosystems are massive. If you're doing NLP, computer vision, or deep learning, Python is *the* language.

Even OpenAI's APIs and LangChain pipelines are Python-first.

I recently built a custom embedding search engine for a client — just 50 lines of Python, and it blew their minds. That kind of magic? Still Python's territory.

## **3. Glue Code**

Python is an amazing “glue” language.

I've used it to:

- Fetch data from REST APIs
- Clean it up with Pandas
- Upload it to a database
- Schedule it with Airflow

It connects the dots like no other language I know. Especially with tools like `requests`, `sqlalchemy`, and `pydantic`, you can build tiny ETL scripts that just work.

## 19 Python One-Liners That Will Transform Your Coding

These small tricks saved me hours. They can do the same for you.

[python.plainenglish.io](https://python.plainenglish.io)

## The Quiet Revolution: Other Tools Doing Data Science Better

I didn't notice it at first. But slowly, the tools around Python got stronger — and more tempting.

### DuckDB

It's like SQLite for analytics. I can query a CSV with SQL in seconds:

```
SELECT COUNT(*), country FROM 'data.csv' GROUP BY country
```

No loading into Pandas. No memory issues. Just results.

And it runs locally — fast. I now reach for DuckDB instead of Pandas more often than I expected.

### Streamlit & Hex

Streamlit was a game-changer for me. I used to spend days turning Jupyter Notebooks into web apps. Streamlit? 10 minutes.

```
import streamlit as st

st.title("Customer Churn Dashboard")
st.line_chart(df["churn_rate"])
```

Recently I've been using [Hex](#), which blends notebooks, SQL cells, and visual UI beautifully. It feels like what Jupyter *wanted* to be.

## **PySpark (Still a Headache, but Useful)**

PySpark was never fun — let's be honest. But for truly massive data (terabytes, not gigabytes), it still matters. I've had to use it for big ETL pipelines, and while it's clunky, it scales.

Python lets me write PySpark, but more often I wish I could skip Python and just write Spark SQL.

## **What I Tell New Data Scientists Now**

People often ask me: “Should I still learn Python for data science?”

My answer?

| *Yes — but don't stop there.*

Learn Python to get your bearings. Build projects, write messy code, have fun.

Then — learn SQL like it's your first language. Pick up a modern tool like dbt or Polars.

Play with Streamlit. Learn enough Bash and Git to get around. Keep an eye on Julia and Rust. And try to solve problems without code, too — just to learn how others think.

Most of all, learn to choose tools *on purpose*, not out of habit.

## **The Future? It's Multilingual**

Here's my honest belief: the best data scientists in 2025 and beyond won't be Python specialists. They'll be *multilingual problem-solvers*.

They'll know:

- SQL like a second language
- Python for custom logic
- A sprinkle of Rust or Julia for performance
- How to use — and not fear — low-code tools
- When *not* to build, but to plug in something that already works

And most importantly? They'll be humble enough to pick the right tool for the job, not the one they're emotionally attached to.

## Wrapping Up

Python's not dead. It's not even in trouble. But it's no longer the top choice for *every* data science task — and that's okay.

It's like an old friend who helped you grow, and now you're both walking different paths. I still visit often. I still use it when I need it. But I've learned to appreciate the bigger toolbox out there.

If you're a new data scientist: learn Python, yes. But don't stop there.

If you're experienced: take a breath, look around. You don't have to do everything in Jupyter.

There's a whole new world of tools waiting to make your life easier.

And honestly? It's exciting.

## Top 10 Python Libraries for Data Visualization That I Use Regularly

These tools helped me create everything from quick charts to full dashboards — without writing much code.

[python.plainenglish.io](https://python.plainenglish.io/)

## Top 50 Python Coding Interview Questions in 2025

A curated guide from someone who's been there, solved that.

[python.plainenglish.io](https://python.plainenglish.io/)

## Thank you for being a part of the community

*Before you go:*

- Be sure to clap and follow the writer 
- Follow us: [X](#) | [LinkedIn](#) | [YouTube](#) | [Newsletter](#) | [Podcast](#) | [Differ](#) | [Twitch](#)
- [Check out CoFeed, the smart way to stay up-to-date with the latest in tech](#) 
- [Start your own free AI-powered blog on Differ](#) 
- [Join our content creators community on Discord](#) 
- For more content, visit [plainenglish.io](#) + [stackademic.com](#)



121



2



Python

Python Programming

Programming

Artificial Intelligence

Technology

[Follow](#)

## Published in Python in Plain English

47K Followers · Last published 1 day ago

New Python content every day. Follow to join our 3.5M+ monthly readers.

[Follow](#)

## Written by PURRFECT SOFTWARE LIMITED

2.5K Followers · 2.1K Following

Learning and Sharing Knowledge Everyday | ❤️ #python #flutter #ai 🍉

## Responses (2)



To respond to this story,  
get the free Medium app.

[Open in app](#)

Andrea Moro  
4 hours ago

[...](#)

Not sure how to say, but claiming not to use Python as the first choice when every sample is in fact implying it's usage sounds a bit counterintuitive.

Not even sure such an article has been written with an AI.. Even a normal AI would have known Python is still there in all you suggested.

👏 21    💬 1 reply

👤 ★ Bruce Hamilton  
10 mins ago

...

As someone who has been on the software management side more recently than the software developer side, I would like to know when a dev can find time to get competent in these other tools? Do you expect the projects to absorb the cost, or should the... [more](#)

👏

More from PURRFECT SOFTWARE LIMITED and Python in Plain English



 In Python in Plain English by PURRFECT SOFTWARE LIMITED

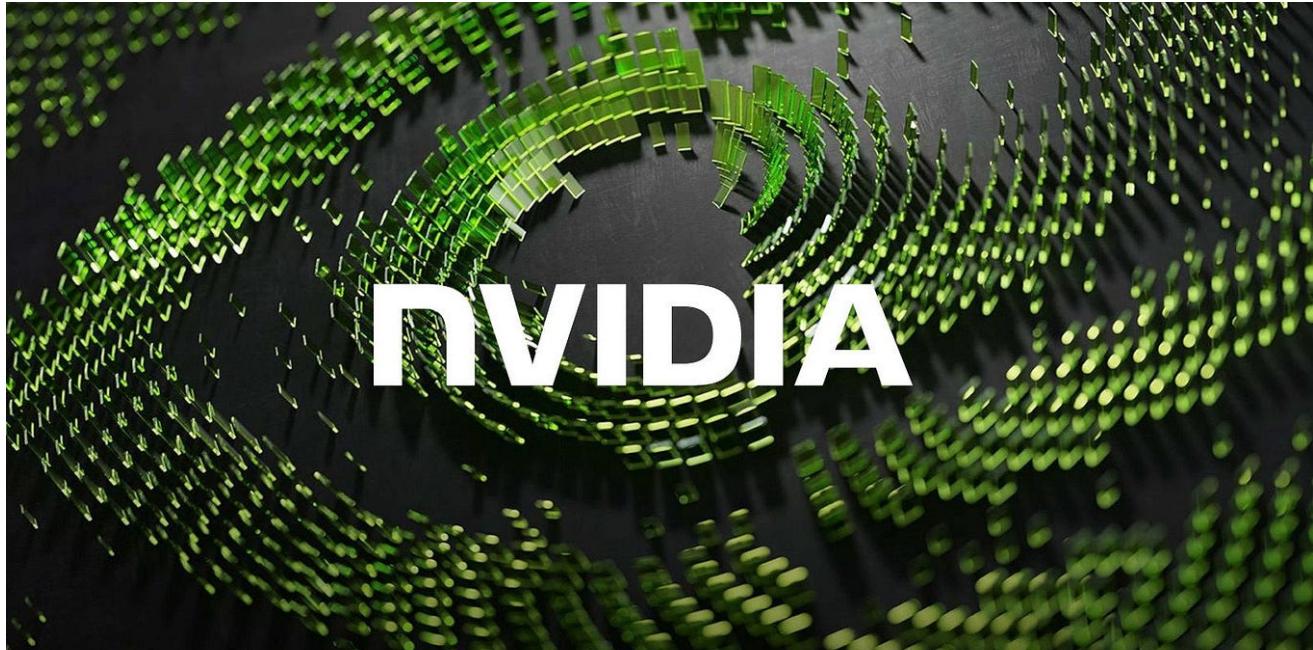
## 10 Uncommon but Extremely Useful Python Libraries

I've been using these underrated Python libraries to work faster, debug more easily, and build smarter tools.

◆ Apr 9  96



...



 In Python in Plain English by Dylan Cooper

## A New Era for GPU Programming: NVIDIA Finally Adds Native Python Support to CUDA—Millions of...

This year, NVIDIA is going all in, making it clear that Python is to become a first-class citizen in its CUDA parallel programming...

◆ Apr 17  731  4



...

```
from functools import partial
```

```
def multiply(x, y):  
    return x * y
```

```
multiply_by_five = partial(multiply, 5)
```

```
multiply_by_five(10)
```

```
50
```



In Python in Plain English by Jaume Boguñá

## 97% of Python Projects Could Be Cleaner With `partial()`

How one built-in tool can save you lines of code and mental energy.

4 Apr 17 976 12



...



 In Python in Plain English by PURRFECT SOFTWARE LIMITED

## 9 Python Libraries That Solved Real Problems in My Data Projects

Simple Python tools that fixed annoying data problems I faced every day.

⭐ Apr 17 🌐 83 💬 1



[See all from PURRFECT SOFTWARE LIMITED](#)

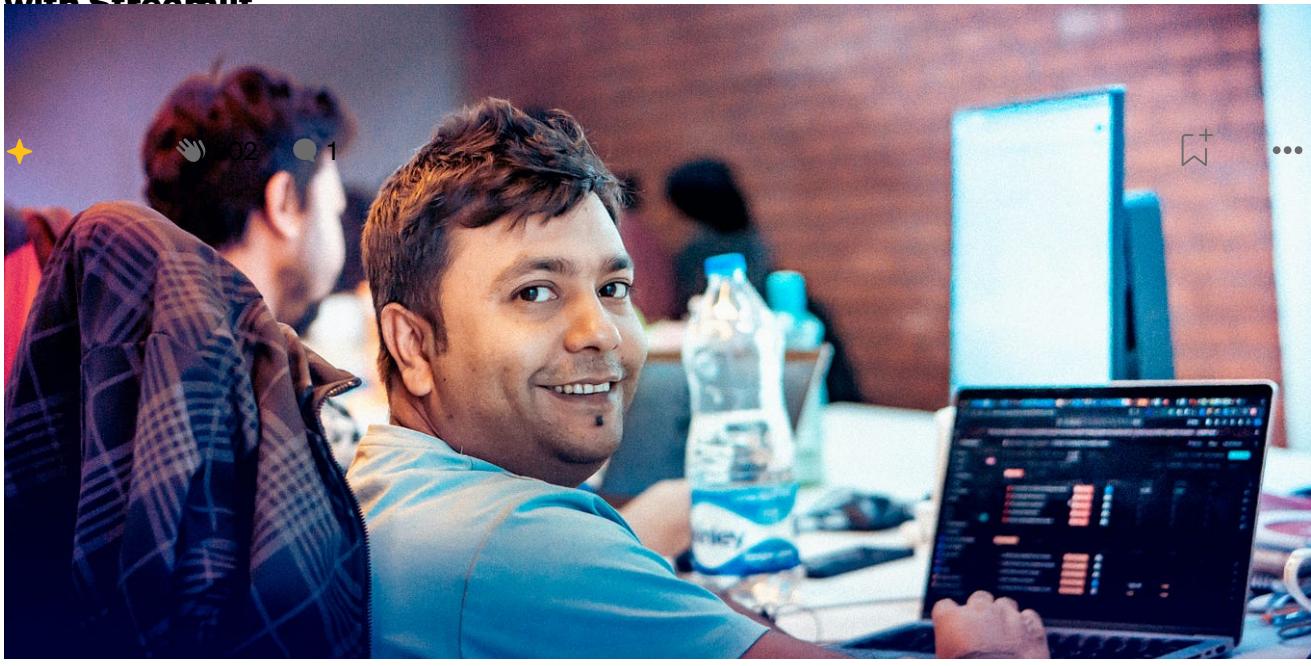
[See all from Python in Plain English](#)

### Recommended from Medium



DSC In Data Science Collective by Raphael Schols

## Let Users Talk to Your Databases: Build a RAG-Powered SQL Assistant with Streamlit



Analyst Uttam

### This SQL Trick Cut My Query Time by 80%

The Problem That Wasted My Hours (And Sanity)

Apr 22 614 33



 In The Pythonworld by Pythonworld

## 5 Modern Python Tools Every Developer Should Master in 2025

Master 5 modern Python tools that will boost your productivity and efficiency in 2025.

 6d ago  259  1



...



 In ILLUMINATION by Dr. Nikita Singh

## After Age 40, You Need to Stop Doing These Morning Habits—That Speed Up Aging

Your Morning Routine Making You Age Faster?

 Apr 21  5.1K  96



...

```
role = input("Enter your role (admin, editor, viewer):")  
match role:  
    case "admin":  
        print("You have full access to the system.")  
    case "editor":  
        print("You can modify content.")  
    case "viewer":  
        print("You can view content.")  
    case _:  
        print("Invalid role")
```



In Python in Plain English by Jaume Boguñá

## Still Using if-elif Chains? Let Me Show You a Better Way

Why I now use pattern matching in every Python project.

⭐ Apr 19 ⚡ 288 🗣 4



...



 In Stackademic by Blend Visions

## 5 Best MCP Servers for Effortless Vibe Coding in 2025 🚀

The time has come for you to maximize your coding process through enhanced productivity while gaining remarkable efficiency gains. The...

 6d ago  361  6[See more recommendations](#)