Python Tools to Deploy Your Machine Learning Models Faster

Jeff Hale

gradio

Streamlit

FastAPI

Welcome Aboard!

Pilot: Jeff Hale

- linkedin.com/in/-jeffhale/
- twitter@discdiver
- jeffhale.medium.com/

Preflight check

What's one thing you're hoping to learn today?

Planes

gradio

Streamlit

FastAPI

Flight plans

- Test flights for each plane
  - Hello world
  - Show data
  - Plot
  - Model inference
- Cruising altitude & turbulence (pros & cons)
- Grab your luggage (takeaways)
- Disembark

Gradio

Ultralight

New, quick to fly, experimental

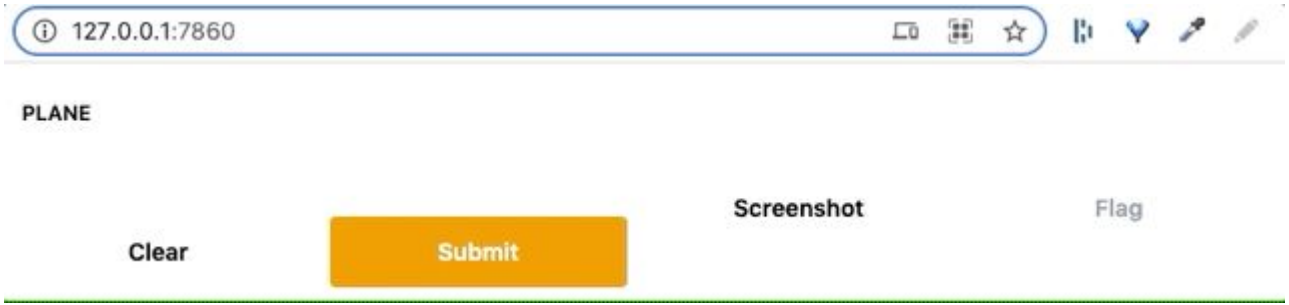
Gradio Demo 1: Hello world

Gradio #1: Hello world

```
import gradio as gr


def hello(plane):
    return f"I'm an ultralight {plane} 🛩️"


iface = gr.Interface(
    fn=hello,
    inputs=['text'],
    outputs=['text']
).launch()
```

Gradio #1: Hello world

- *pip install gradio*
- *python gradio_hello.py*



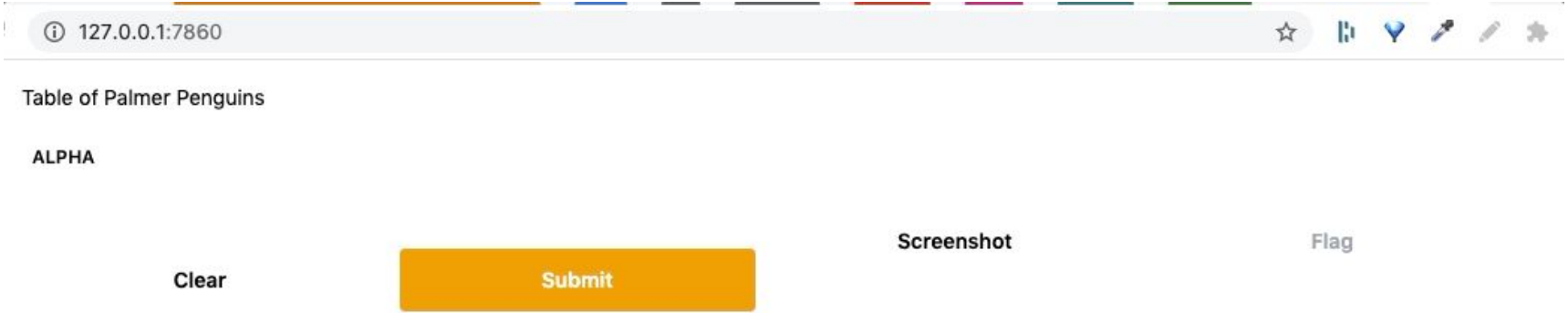
Gradio Demo 2: Show data

Gradio #2: Show me the data!

```
import gradio as gr
import pandas as pd

def show_pens(alpha):
    return pd.read_csv(
      'https://raw.githubusercontent.com/mwaskom/seaborn-data/master/penguins.csv')

iface = gr.Interface(
    fn=show_pens,
    inputs=['text'],
    outputs=[gr.outputs.Dataframe()],
    description="Table of Palmer Penguins"
).launch(share=True)
```

Gradio #2: Show me the data!

- *pip install gradio pandas seaborn*
- *python gradio_pandas.py*
- Glitch only allows 10 cells to show initially or fails silently as of Nov. 16, 2021 TK



Gradio Demo 3: Plotting

Gradio #3: Plot it

- Plotly doesn't work ☹️
- You can use Matplotlib in theory, but it wouldn't work when I tried Nov. 2021

TK try now

Gradio Demo 4: Model inference

Gradio #4: Model inference

```
import gradio as gr

gr.Interface.load('huggingface/gpt2').launch()
```

Gradio #4: Model inference - prettier

```
gr.Interface.load(
    "huggingface/gpt2",
    title="Storytelling with GPT2",
    css="""
        body {background: rgb(2,0,36);
              background: linear-gradient(
                180deg,
                rgba(2,0,36,1) 0%,
                rgba(7,51,99,1) 70%,
                rgba(6,3,17,1) 100%);}
        .title {color: white !important;}
        .article {color: white !important; font-size: 1.3em;}
        """,
).launch()
```

Gradio #4: Model inference - prettier

Gradio Data API - One Click!

Response:

```
{
  "data": [ Union[str, number] ],
  "durations": [ float ], // the time taken for the prediction to complete
  "avg_durations": [ float ] // the average time taken for all predictions so far (used to estimate the runtime)
}
```

Try it (live demo):

Python | cURL | Javascript

```
curl -X POST http://127.0.0.1:7860/api/predict -H 'Content-Type: application/json' -d '{"data": ["Hello World"]}'
```

Gradio - Hugging Face Spaces Integration

+

Abubakar Abid

Gradio Pros

- Quick demos for ML 🚀
- Built-in interpretability 🔎
- Auto-docs 📃
- Nice Hugging Face integration 🤗
- Bright future 😎

Gradio Cons

- Rough around the edges - early stage 🔪
- Not easy to customize style ✨
- Single page only 1️⃣

Streamlit

Cessna Citation Longitude

Light, quick to takeoff, easy flying

Streamlit

Streamlit Demo 1: Hello world

Streamlit #1: Hello world

```
import streamlit as st

name = "Jeff"

st.title(f"Hello from Streamlit, {name}!")
```

- *pip install streamlit*
- *streamlit run streamlit_hello.py*

Streamlit

Streamlit Demo 2: Show data

Streamlit #2: Show data

```
import streamlit as st
import pandas as pd

st.title("Streamlit with pandas")
show = st.checkbox("Show dataframe")
df_pens = pd.read_csv(
"https://raw.githubusercontent.com/mwaskom/seaborn-data/master/penguins.csv")

if show:
   df_pens
```

Streamlit #2: Show data

- *pip install streamlit pandas seaborn*
- *streamlit run streamlit_pandas.py*

Streamlit #2: Show data

Streamlit

Streamlit Demo 3: Plotting

Streamlit #3: Plotting

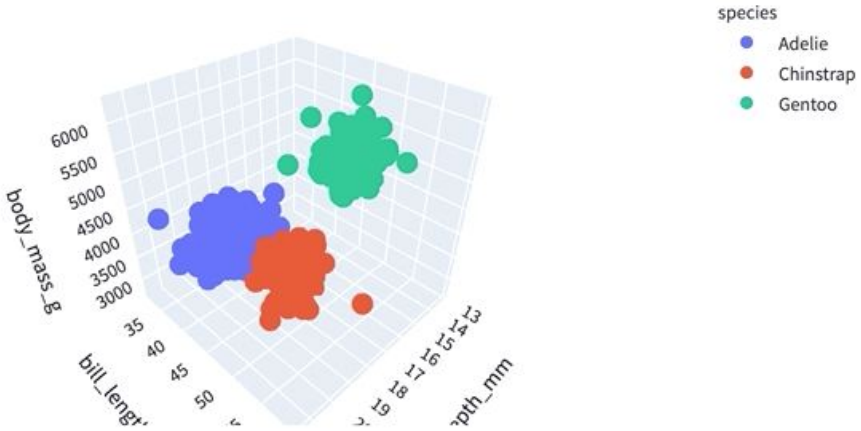
```
choice = st.radio("Select color", ["species", "island"])


fig = px.scatter_3d(
    data_frame=df_pens,
    x="bill_depth_mm",
    y="bill_length_mm",
    z="body_mass_g",
    color=choice,
    title="Penguins in 3D!",
)
fig
```

Streamlit #3: Plotting

Select color

- species
- island

Penguins in 3D!



Streamlit

Streamlit Demo 4: Model Inference

Streamlit #4: Model inference

```
import streamlit as st
from transformers import pipeline


st.header("GPT-2 Stories")
input_text = st.text_area("Enter your text here:")


generator = pipeline("text-generation", model="gpt2")
output = generator(input_text, max_length=100, )
output[0]["generated_text"]
```

Streamlit #4: Model inference

GPT-2 Stories

Enter your text here:

ok friends, let's talk

ok friends, let's talk

Riot - I'm on the board now. Please explain your message.

No one likes news on my channel.

Please put your email in the contact form. - Yes I have a problem

Streamlit #4: Model inference - prettier

GPT-2 Stories

Enter your text here:

ok friends, let's talk

ok friends, let's talk

Riot - I'm on the board now. Please explain your message.

No one likes news on my channel.

Please put your email in the contact form. - Yes I have a problem

Streamlit #4: Model inference - prettier

```
st.header("Story time")
st.image("https://cdn.pixabay.com/photo/2017/07/12/19/03/highway-2497900_960_720.jpg")
col1, col2 = st.columns(2)

with col1:
    input_text = st.text_area("Enter your text here:")
    with st.spinner("Generating story..."):
        generator = pipeline("text-generation", model="gpt2")
        if input_text:
            generated_text = generator(input_text, max_length=60)
            st.success("Here's your story:")
            generated_text[0]["generated_text"]

with col2:
    st.header("I'm in another column")
```

Streamlit #4: Model inference - prettier

Story time

Enter your text here:

ok smarty, here we are

I'm in another column

Here's your story:

ok smarty, here we are.

There's also the recent news of IBM's (IBM) (IBM Inc) (IBM) (IBM) (IBM MS) (IBM) smartwatches and the new BlackBerry (R) in this week's Best

Streamlit Serving Options

- Serve from Streamlit's servers for free. Example: bit.ly/st-6plots
- Or serve from Hugging Face or Heroku for free.
- Or pay Streamlit for more/better
- Or host elsewhere

Streamlit Pros

- Quick websites for many Python use cases 🐍
- Many intuitive interactive widgets ✅
- Nice hosting options 🎁
- Thoughtful docs 📄
- Strong development cadence & team 💪

Streamlit Cons

- Some customizations ✨
- Single page only 1️⃣

FastAPI

Boeing 737

Commercial grade, fast, smart!

FastAPI Demo 1: Hello world

FastAPI #1: Hello world

```
import uvicorn
from fastapi import FastAPI

app = FastAPI()

@app.get("/")
def home():
    return {"Hello world": "How's it going?"}

if __name__ == "__main__":
    uvicorn.run("fastapi_hello:app", reload=True)
```

FastAPI #1: Hello world

pip install fastapi uvicorn

python fastapi_hello.py

Returns json

FastAPI: Async

FastAPI #1: Hello world

```python
@app.get("/not-async")
def home():
    return dict(zip(range(10), range(10)))
```

10ms

FastAPI #1: Hello world

```
@app.get("/yes-async")
async def home():
    return dict(zip(range(10), range(10)))
```

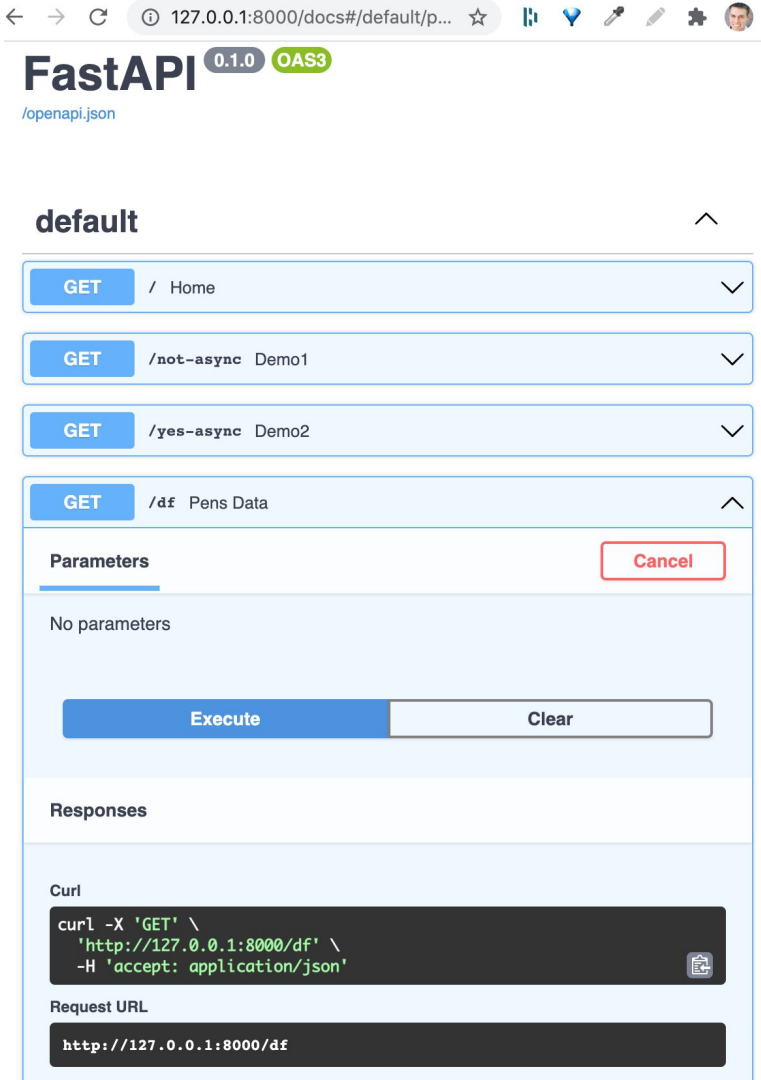
3ms

FastAPI Demo 2: Show data

FastAPI #2: Show me the data!

```
@app.get("/df")
async def pens_data():
    df_pens = pd.read_csv(
     "https://raw.githubusercontent.com/mwaskom/seaborn-data/master/pe
     nguins.csv")
    df_no_nans = df_pens.fillna(-1.01)
    return df_no_nans
```

Automatic docs

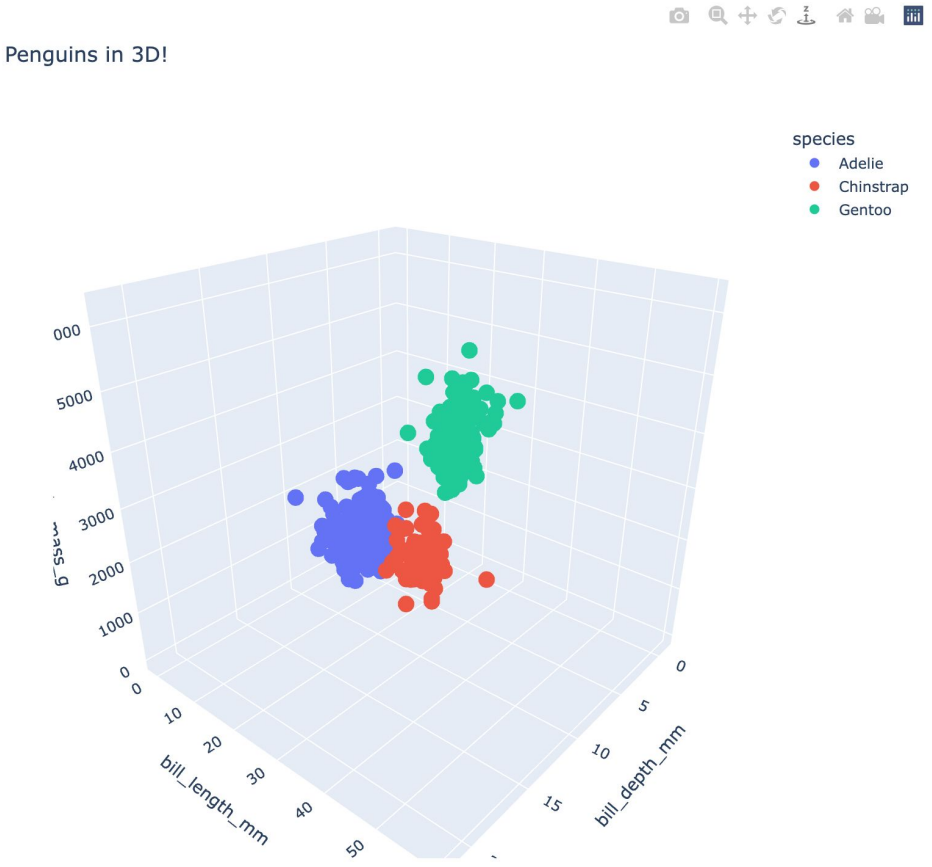


FastAPI Demo 3: Plotting

FastAPI #3: Plotting

```
from fastapi.responses import HTMLResponse
import plotly.express as px
...
@app.get("/plot")
async def plot() -> HTMLResponse:
    """return a plotly plot"""
    fig = px.scatter_3d(
        data_frame=df,
        x="bill_depth_mm",
        y="bill_length_mm",
        z="body_mass_g",
        color="species",
        title="Penguins in 3D!",
    )
    return HTMLResponse(fig.to_html())
```

FastAPI #3: Plotting



FastAPI Demo 4: Model Inference

FastAPI #4: Model inference

Typing with mypy TK

FastAPI Pros

- Fastest Python API framework - async 🏎️
- Automatic API documentation ⭐
- Nice error messages ⚠️
- Extensive docs 📃
- Jinja templating ⚔️
- Nice test client 👌
- SQL Model integration ↔️

FastAPI Pros

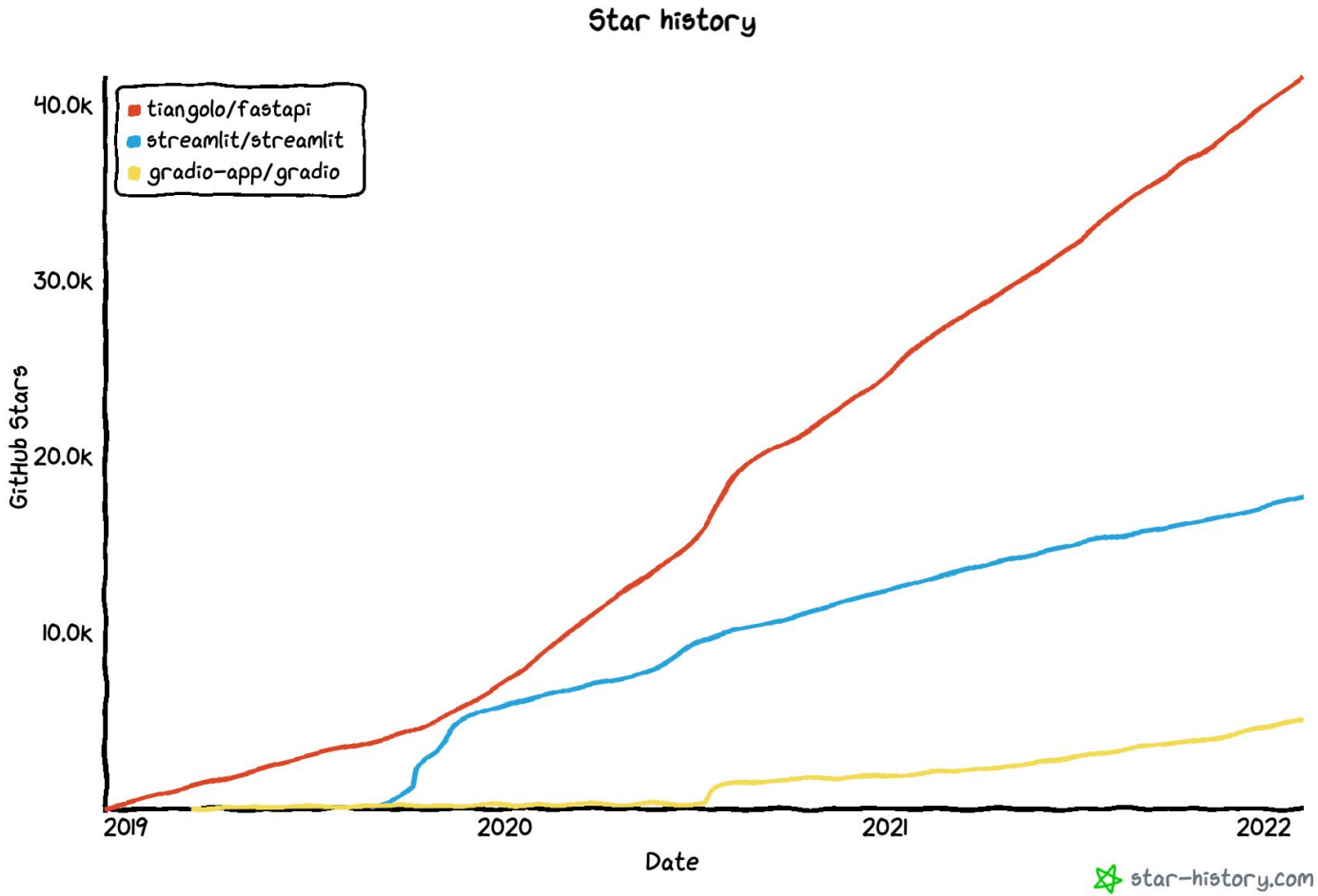
Sebastian Ramirez

FastAPI Cons

- Reliant on a single maintainer 😬
- Takes more code for HTML templating than Flask ⌨️

Summin' Up

	Web App	**Data API**
gradio	Yes	Yes
Streamlit	Yes	No
FastAPI	Yes (Jinja templates)	Yes



Source: https://star-history.t9t.io/#streamlit/streamlit&tiangolo/fastapi&gradio-app/gradio

Grab Your Luggage (takeaways)

Use what you know, unless it doesn't meet your needs.

Blank slate?

Learn what's popular, growing, and quick to get off the ground.

Streamlit

For single-page app that doesn't need custom styling.

Gradio for quick 🤗 models for fun.

FastAPI for serving data.

What to learn next?

(Newer to Python)

Streamlit

Disembark

Thank you for flying the deployment skies!

Jeff Hale

- linkedin.com/in/-jeffhale/
- @discdiver
- jeffhale.medium.com/

Versions used

- Gradio 2.4.5
- Streamlit 1.0.2
- FastAPI 0.70

Python Tools to Deploy Your Machine Learning Models Faster

Jeff Hale

gradio

Streamlit

FastAPI