

Getting Started with Streamlit

February 2022



Streamlit

Outline

- Introduction to Streamlit (20 mins)
 - Why Streamlit exists
 - Streamlit outputs: markdown, “magic”, displaying dataframe, graph
 - Streamlit inputs: using widgets for interactive apps
 - Laying out widgets
 - Streamlit Components: extending what’s possible with Streamlit
- In-depth demo (30 mins)
- Where to get more information and Q&A (10 mins)

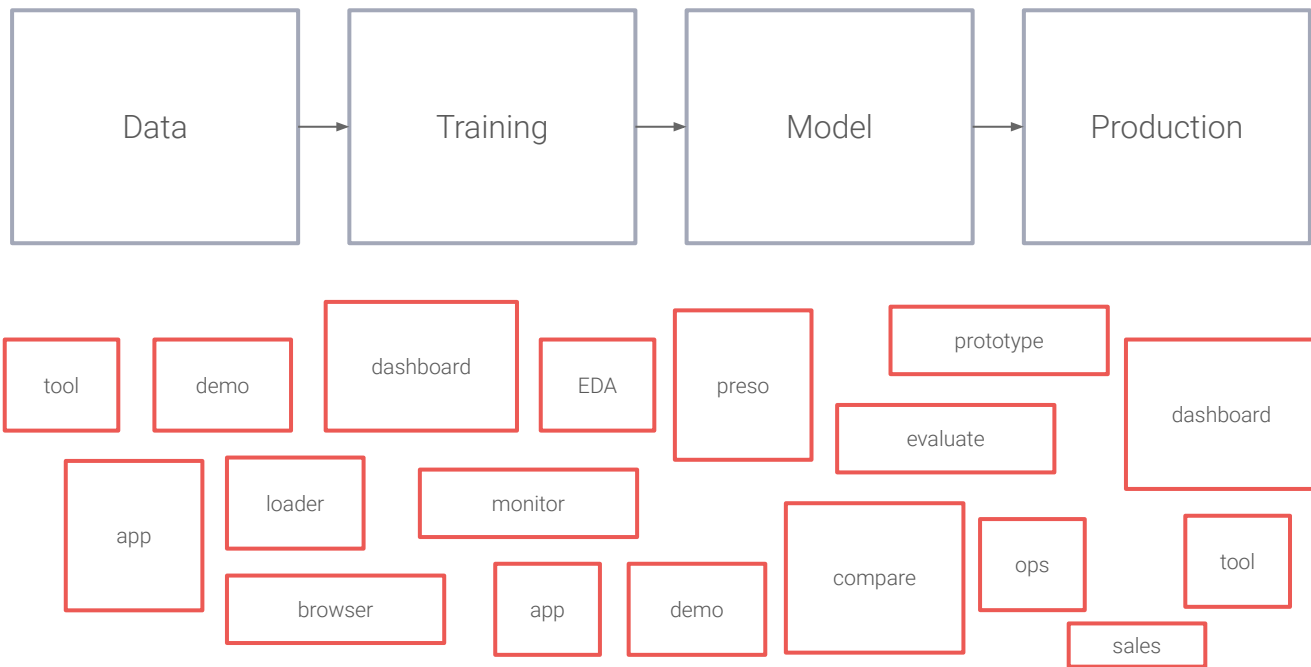


Introduction to Streamlit



What **apps** do data scientists need?

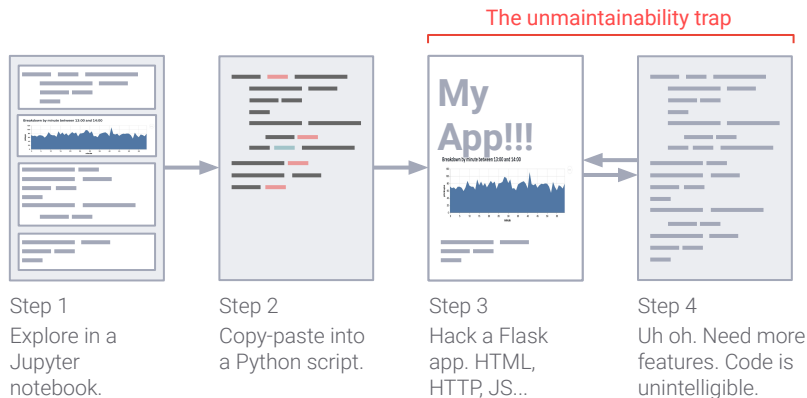
Lots! You're always making apps for your work or to communicate your results to others.



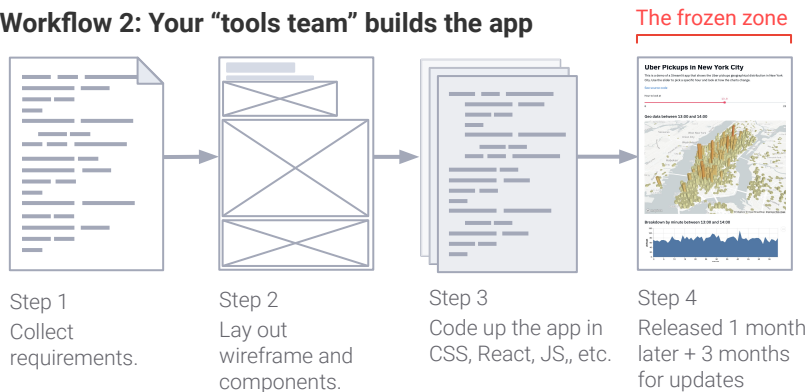
Building data apps is slow and expensive

Building even simple data apps today requires weeks or months of investment, distracts from core work, and often yields an unmaintainable product. And because it's so costly **only a fraction of needed apps and tools are created.**

Workflow 1: You build the app



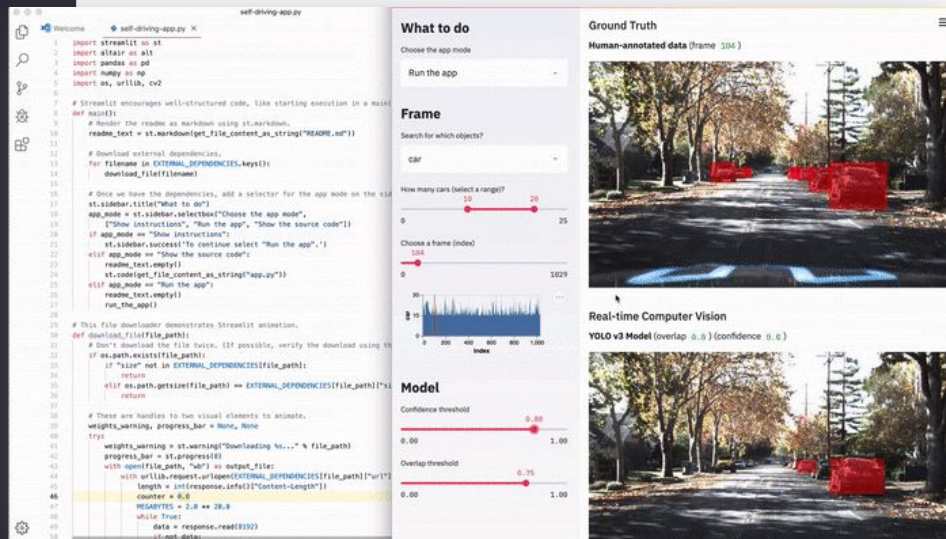
Workflow 2: Your “tools team” builds the app



Streamlit is the fastest and easiest way to create data apps

Quickly create your own elegant data apps for visualization, debugging, comparing models and presenting data - all in Python.

Streamlit's open-source app framework is built specifically for data scientists to rapidly create beautiful, performant apps in only a few hours!



Streamlit works on 3 simple principles



Embrace
Python scripting

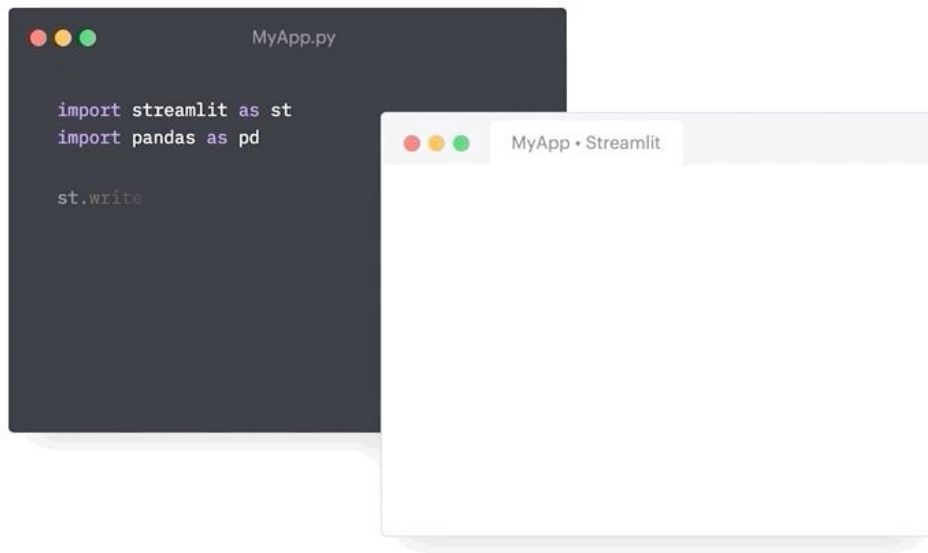


Treat widgets
as variables.

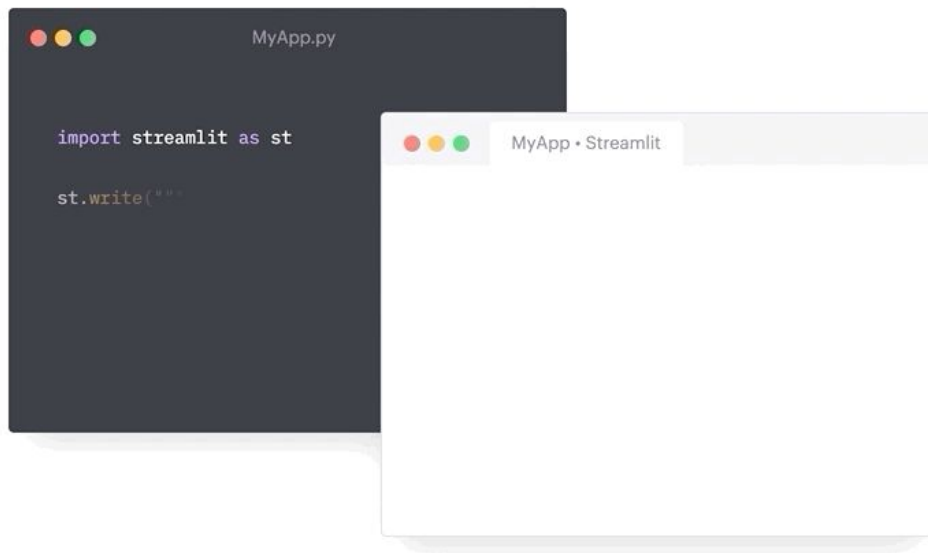


Reuse data and
computation.

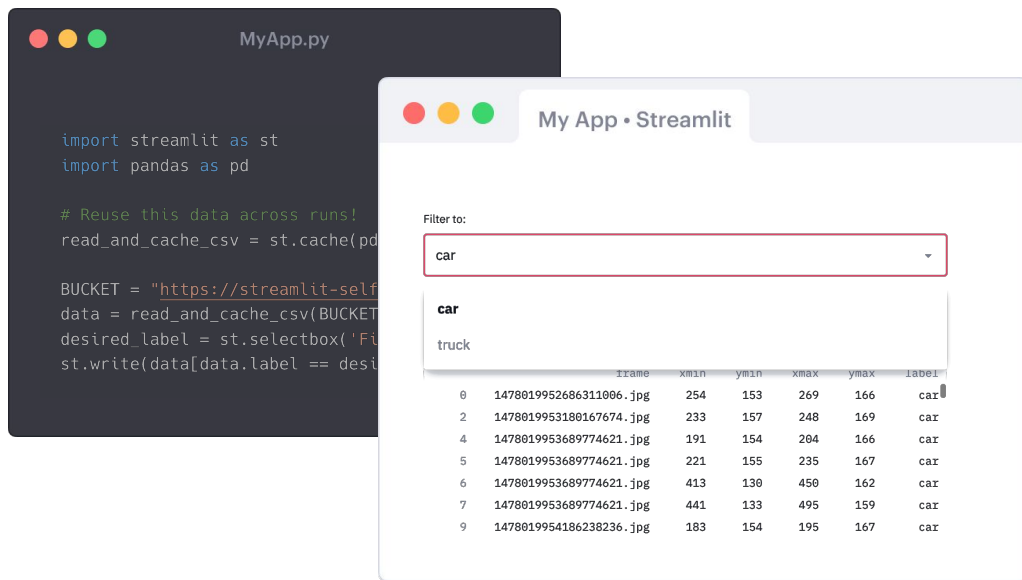
Embrace Python scripting



Treat **widgets** as variables



Reuse data and computation



The image shows a code editor window titled 'MyApp.py' and a Streamlit application window titled 'My App • Streamlit'.

Code Editor (MyApp.py):

```
import streamlit as st
import pandas as pd

# Reuse this data across runs!
read_and_cache_csv = st.cache(pd.read_csv)

BUCKET = "https://streamlit-self-hosted-s3-us-east-1.amazonaws.com/streamlit-demo-data/mini-vehicles.json"
data = read_and_cache_csv(BUCKET)
desired_label = st.selectbox('Filter to:', data['label'].unique())
st.write(data[data.label == desired_label])
```

Streamlit Application (My App • Streamlit):

Filter to:

car

car

truck

		frame	xmin	ymin	xmax	ymax	label
0	1478019952686311006.jpg	254	153	269	166	car	
2	1478019953180167674.jpg	233	157	248	169	car	
4	1478019953689774621.jpg	191	154	204	166	car	
5	1478019953689774621.jpg	221	155	235	167	car	
6	1478019953689774621.jpg	413	130	450	162	car	
7	1478019953689774621.jpg	441	133	495	159	car	
9	1478019954186238236.jpg	183	154	195	167	car	

In-depth Demo

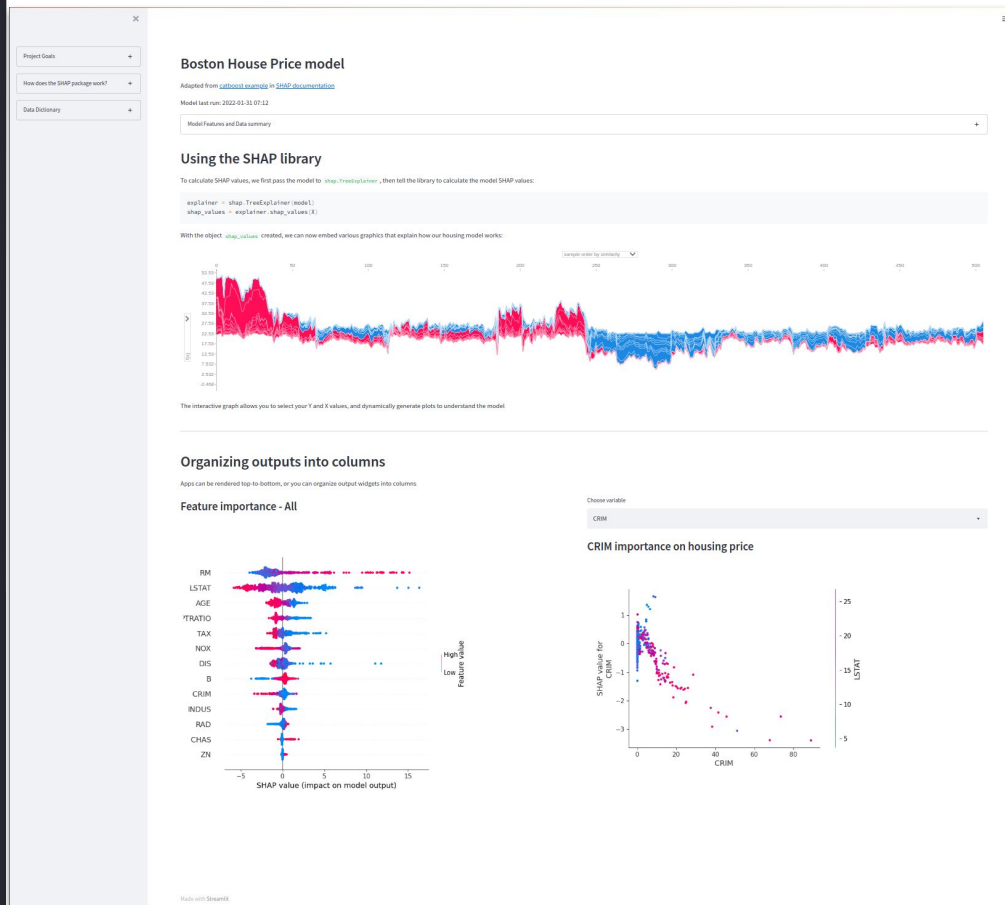


Explaining house pricing model

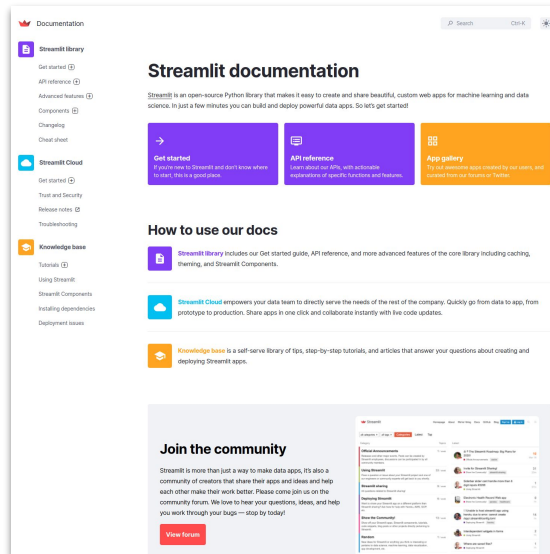
Once you've estimated a model, you need to "sell" that solution to your stakeholders.

Showing is better than just *telling*; here's how you can use the SHAP library in combination with Streamlit to explain why the model predicts the way it does

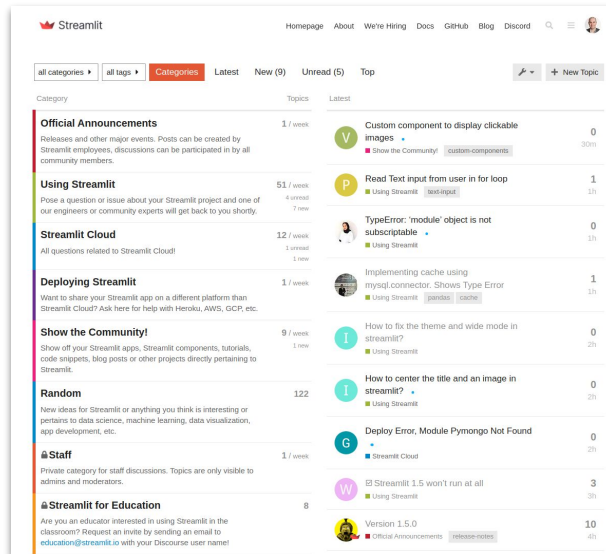
<https://github.com/streamlit/demo-streamlit-shap>



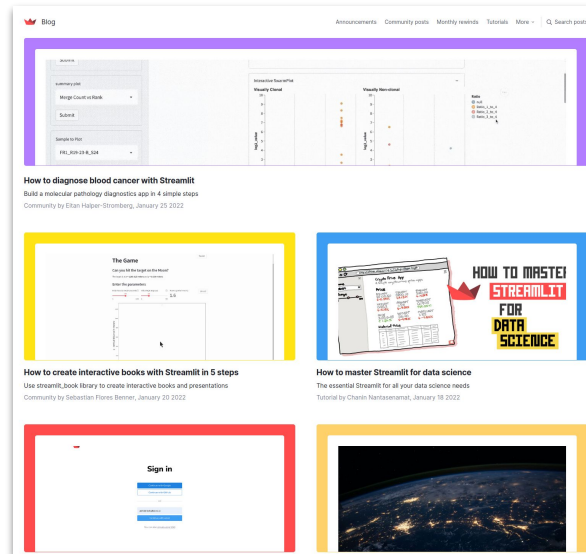
Where to get more information



<https://docs.streamlit.io>



<https://discuss.streamlit.io>



<https://blog.streamlit.io>

