SIC Batch 5

Week 8 - Introduction to Streamlit

Streamlit

Streamlit is a free and open-source framework to rapidly build and share beautiful machine learning and data science web apps. It is a Python-based library specifically designed for machine learning engineers

Streamlit

Why Streamlit?

- **Open-Source Python Library**: Empower your data science and machine learning projects with a robust, open-source Python library.
- **Interactive User Interfaces**: Build intuitive and interactive web interfaces without extensive front-end expertise.
- **Low-Code Development:** Streamline development with pre-built widgets and elements, allowing you to create web pages with minimal coding.
- **Extensive Compatibility**: Seamlessly integrate with your existing Python ecosystem for a cohesive development experience (e.g. pandas, matplotlib, seaborn, plotly, Keras, PyTorch, SymPy(latex)).

Streamlit Getting Started

Install & Import

```
pip install streamlit

streamlit run first_app.py

# Import convention
>>> import streamlit as st
```

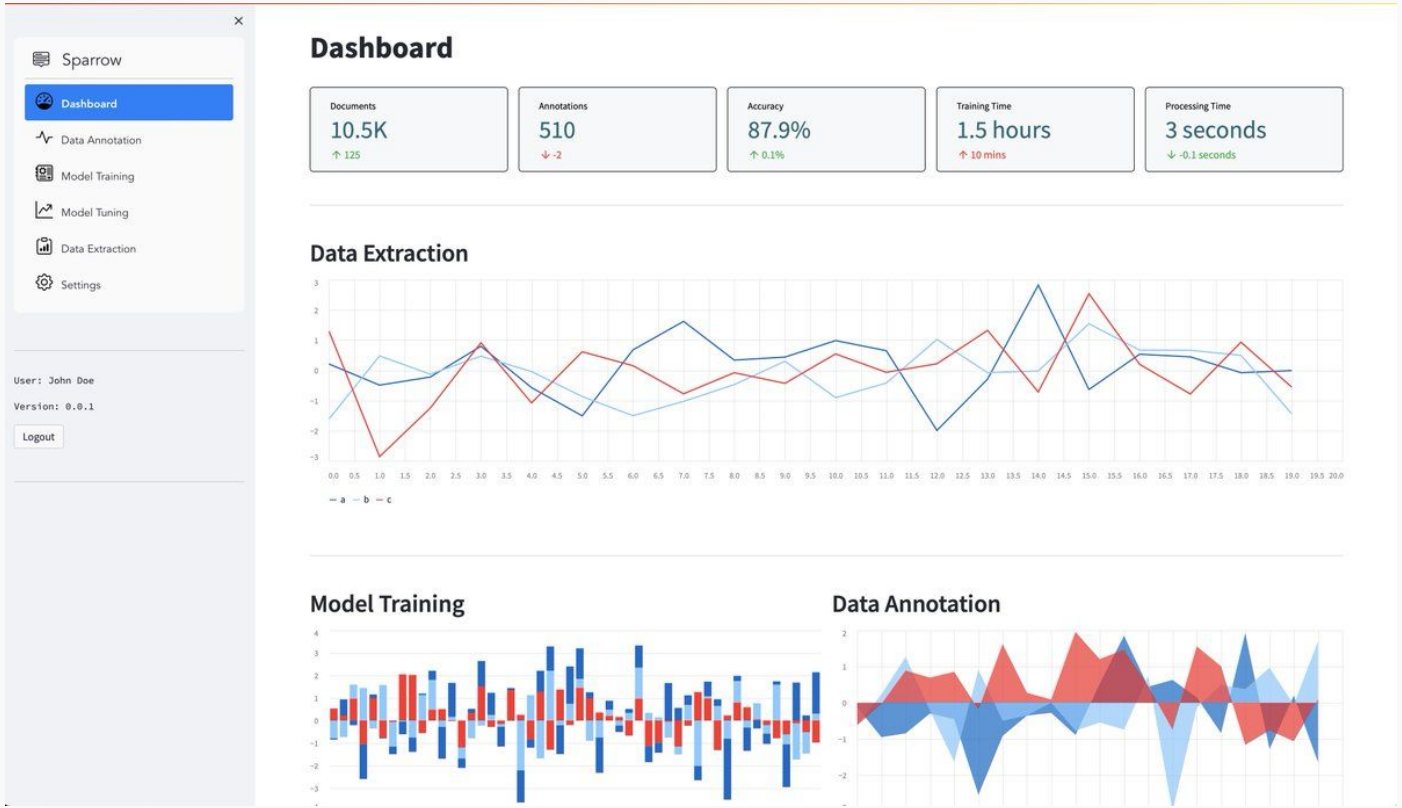
Command Line

```
streamlit --help
streamlit run your_script.py
streamlit hello
streamlit config show
streamlit cache clear
streamlit docs
streamlit --version
```

Streamlit Page Elements & Capabilities

- **Text Elements**
  - Working with text information (Insert text, markdown, title, etc)
- **Data Elements**
  - Working with data, includes dataframe, metrics, JSON, static tables, etc)
- **Chart Elements**
  - Visualizing data through chart (bar, line, scatter, etc)
- **Input Widgets**
  - Interact directly into your apps with buttons, sliders, text inputs, and more
- **Media Elements**
  - Embed images, videos, and audio files directly into Streamlit apps.
- **Layouts and Containers**
  - Controlling how different elements are laid out on the screen
- And many more, read on https://docs.streamlit.io/develop/api-reference

Streamlit

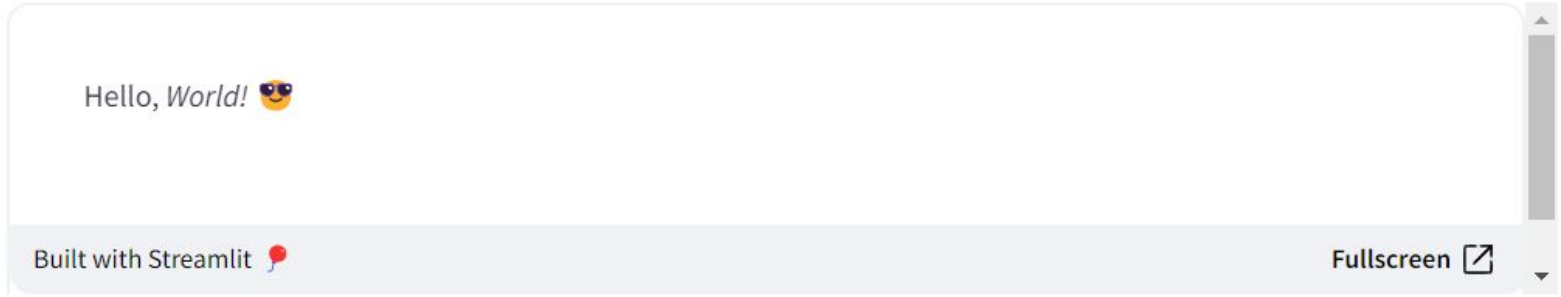

Sparrow
Dashboard
Data Annotation
Model Training
Model Tuning
Data Extraction
Settings
User: John Doe
Version: 0.0.1
Logout
Dashboard
Documents
10.5K
↑ 125
Annotations
510
↓ -2
Accuracy
87.9%
↑ 0.1%
Training Time
1.5 hours
↑ 10 mins
Processing Time
3 seconds
↓ -0.1 seconds
Data Extraction
a b c
Model Training
Data Annotation

Streamlit code

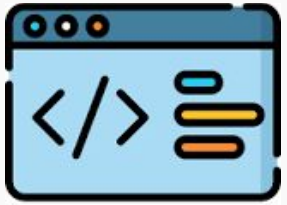
Menampilkan teks pada streamlit

```
import streamlit as st

st.write('Hello, *World!* :sunglasses:')
```



Streamlit code



Menampilkan teks

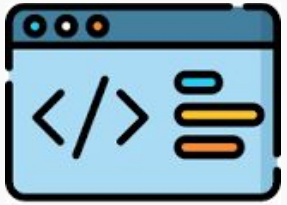
```
st.write("Most objects")
st.write(["st", "is <", 3])
st.write_stream(my_generator)

st.text("Fixed width text")
st.markdown("_Markdown_")
st.title("My title")
st.header("My header")
st.subheader("My sub")
st.code("for i in range(9):
do()")
```

Menampilkan data

```
st.dataframe(my_dataframe)
st.table(data.iloc[0:10])
st.json({"foo":"bar","fu":"ba"})
st.metric("My metric", 42, 2)
```

Streamlit code



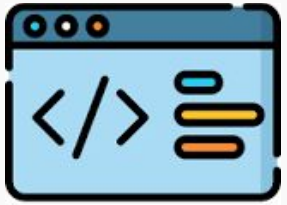
Menampilkan media

```
st.image("./header.png")
st.audio(data)
st.video(data)
st.video(data,
subtitles="./subs.vtt")
```

Membuat tab

```
# Menambahkan tab:
>>> tab1, tab2 = st.tabs(["Tab
1", "Tab2"])
>>> tab1.write("Ini tab 1")
>>> tab2.write("Ini tab 2")
```

Streamlit code

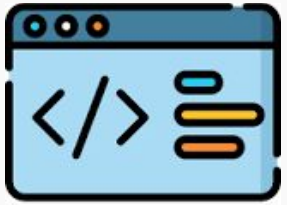


Menampilkan grafik

```
st.area_chart(df)
st.bar_chart(df)
st.line_chart(df)
st.map(df)
st.scatter_chart(df)
```

```
st.altair_chart(chart)
st.bokeh_chart(fig)
st.graphviz_chart(fig)
st.plotly_chart(fig)
st.pydeck_chart(chart)
st.pyplot(fig)
st.vega_lite_chart(df)
```

Streamlit code

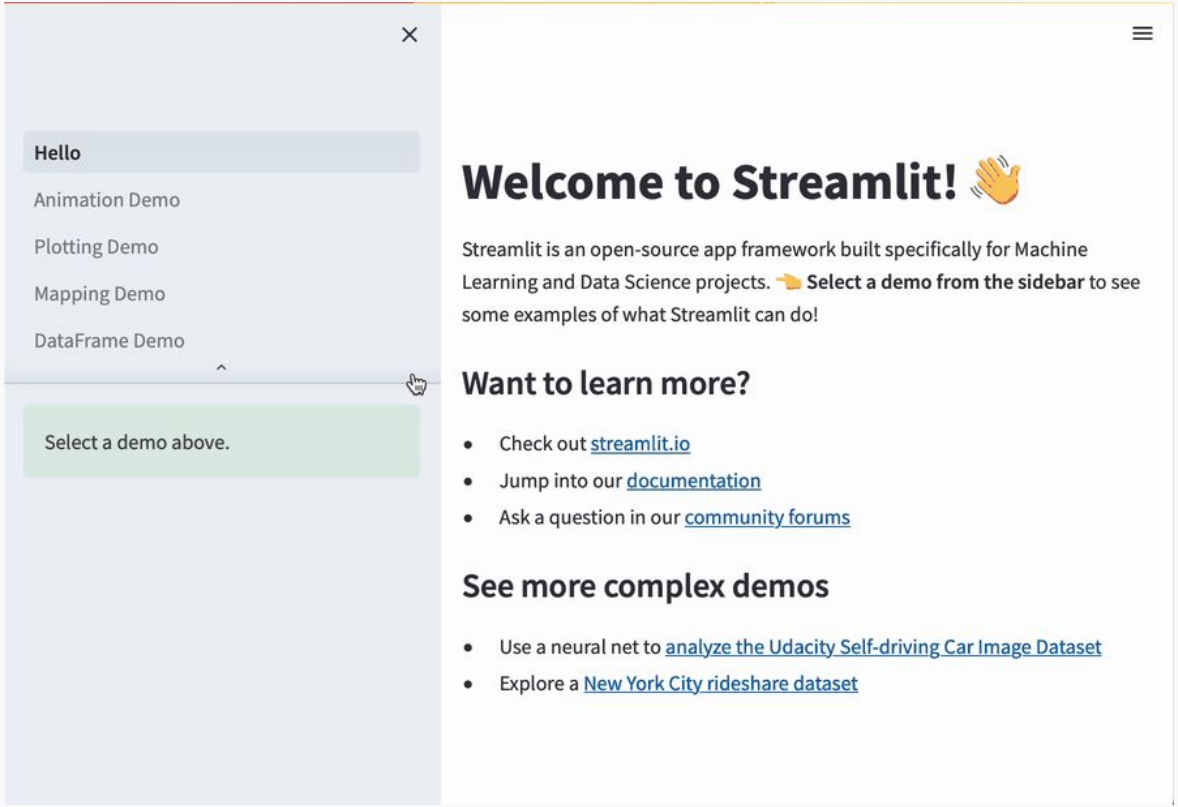


Menampilkan widget interaktif

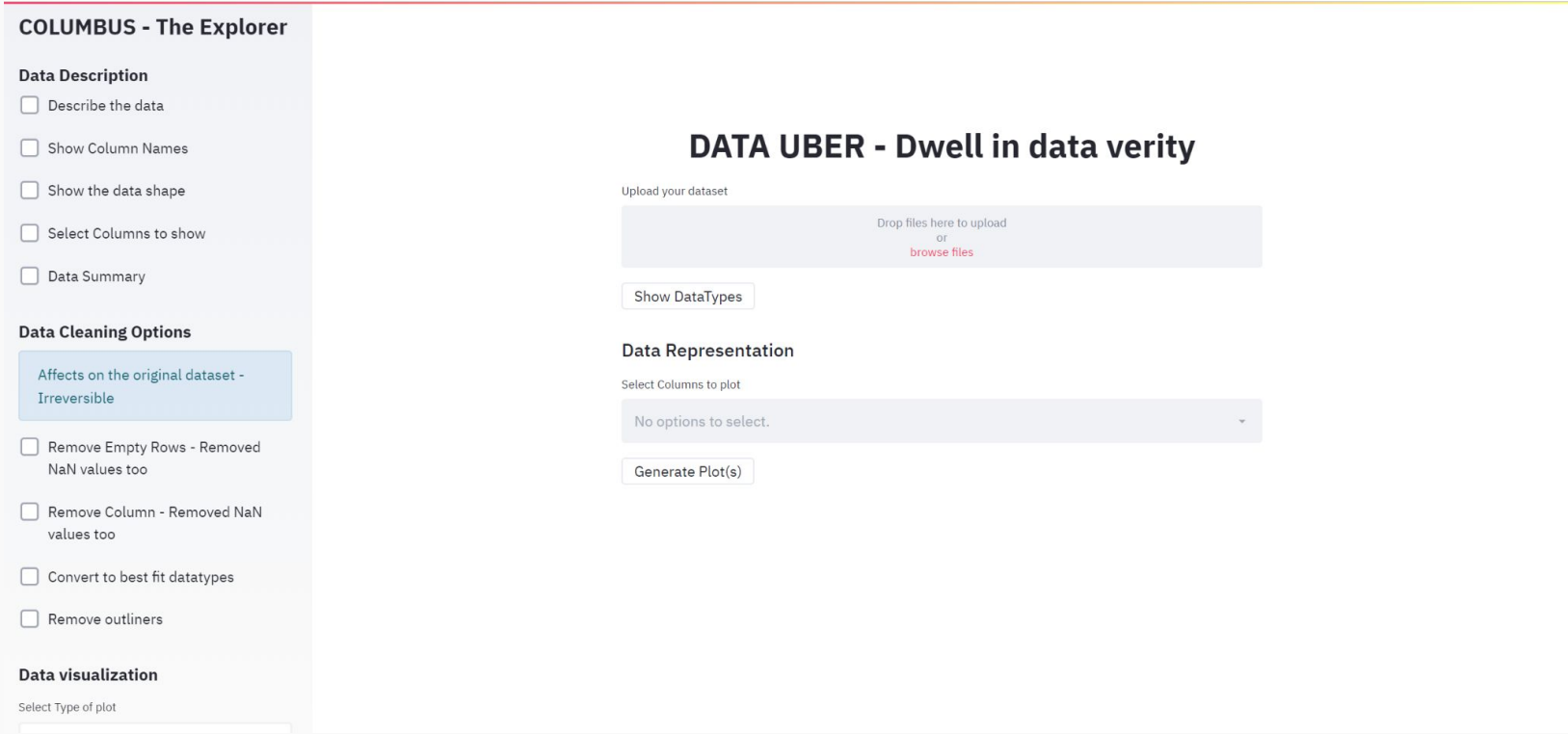
```
st.button("Klik di sini")
st.download_button("Unduh file",
data)
st.link_button("Ke laman utama",
url)
st.page_link("app.py",
label="Home")
st.data_editor("Edit data", data)
st.checkbox("Setuju")
st.toggle("Enable")
```

```
st.radio("Pilih salah satu",
["kucing", "hamster"])
st.selectbox("Pilih salah satu",
["kucing", "hamster"])
st.multiselect("Beli", ["kopi",
"susu", "teh"])
st.slider("Pilih angka", 0, 100)
```

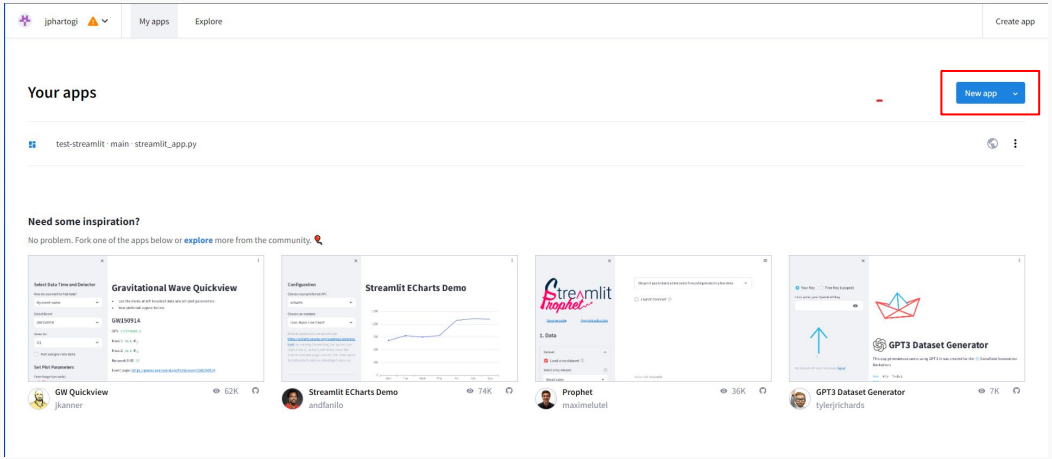
Streamlit Demo App

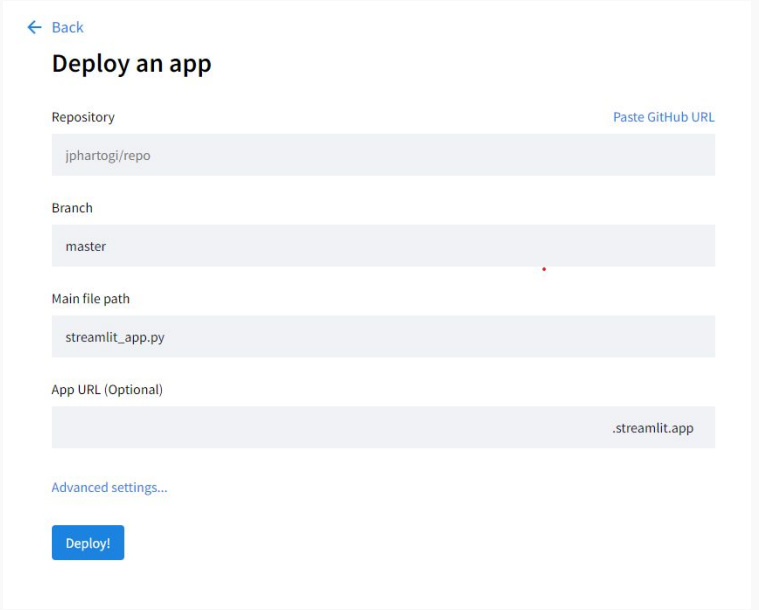


Contoh streamlit app

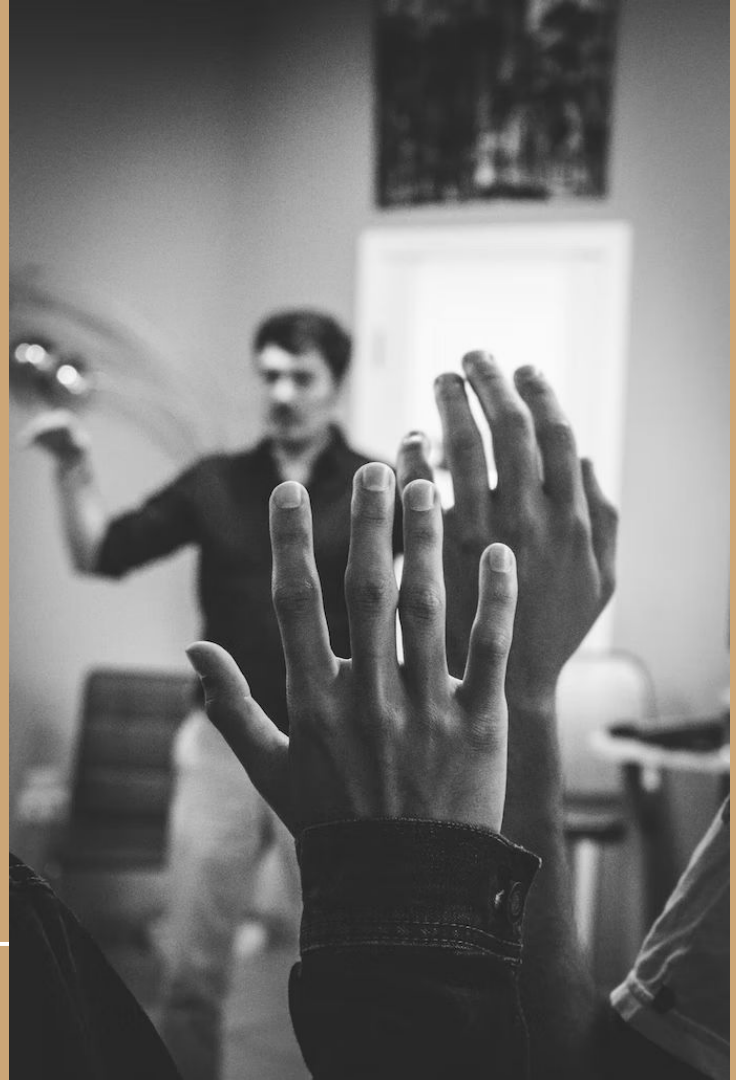


Deploy app on public URL





Demo



Challenge!

Streamlit Challenge:

- Create and improve the UI to your style of choice
- Change the video processing app to image (.jpg, .png) processing app!
- Deploy your app to the cloud so it can be accessed from anywhere!