

# Streamlit 调研报告

## 1 Streamlit 概述

Streamlit 是一个开源的 Python 框架，专为数据科学家和 AI/ML 工程师设计，仅需几行代码即可构建动态数据应用<sup>1</sup>。相比于 Django（发布于 2005 年）和 Flask（发布于 2010 年），Streamlit（发布于 2019 年）的诞生时间较短，技术生态也相对年轻，但是凭借其简洁的语法和专用的功能在最近几年内热度不断飙升（如图 1 所示<sup>2</sup>），成为了一款数据科学和机器学习领域备受喜爱的 pythonWeb 工具。

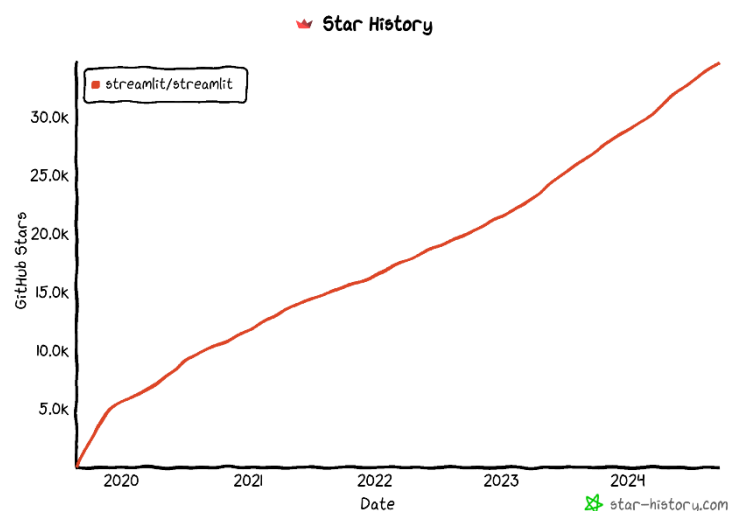


图 1 Streamlit Github stars 数量增长曲线

Streamlit 的主要优点有：

- 简单易用：学习成本低，上手容易，每个了解 python 的人都可以使用 streamlit 在几分钟内用简短的几行代码将数据脚本转换为可共享的 Web 应用，不要求掌握前端知识（HTML、CSS、JavaScript 等）。
- 实时更新：在变换代码时只需刷新浏览器页面即可自动更新应用。
- 交互性强：内置多种常用的 UI 组件（包括复选框、侧边栏、文件上传、进度条），方便用户与系统进行交互。
- 部署便捷：支持多种部署方式，可以实现一键部署分享。

---

<sup>1</sup> Streamlit 官方介绍：[Streamlit documentation](https://docs.streamlit.io/)

<sup>2</sup> 数据来源：[GitHub Star History \(star-history.com\)](https://star-history.com/)

- 专用性强：支持多种专业第三方库，如 Matplotlib、Dagre-d3、Vega-Lite 等可视化库。

## 2 Streamlit 在数据科学中的应用

### 2.1 数据探索

用户可以通过 Streamlit 创建滑块和下拉框等组件动态筛选数据，分别检视数据集的各个子集的情况，还可以观察数据分布、均值、方差、孤立点等统计量。

### 2.2 模型展示

在机器学习中，用户还可以快速将复杂的机器学习模型及其预测结果展示出来。它支持对模型输入、输出以及中间处理步骤进行实时交互，使数据科学家能够轻松地展示模型性能、调试模型行为，并让非技术用户参与到模型的使用和反馈中。Streamlit 的快速迭代和自动刷新功能，提升了模型开发和展示的效率，使其成为机器学习领域不可或缺的展示工具。

图 2 是一个在加利福尼亚房价数据集上训练的线性回归模型通过 Streamlit 在网页上的具体展示<sup>3</sup>，普通用户能使用它得到预测房价大小，而开发者也可以根据特征重要性的可视化展示来进行模型调优。

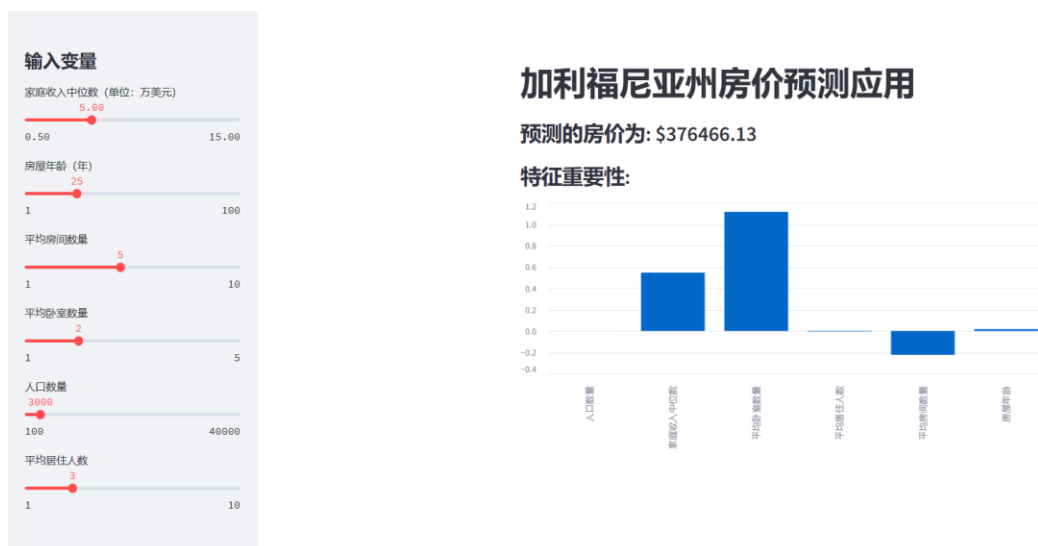


图 2 房价预测应用

<sup>3</sup> 应用代码已上传至 Github，见 [https://github.com/france122/Streamlit\\_learning](https://github.com/france122/Streamlit_learning)

## 2.3 结果可视化

(1) 可视化 DataFrame，对于时序数据绘制图像进行对比分析，如图 3 为对多个国家历史 GAP 数据的展示。

### DataFrame Demo

This demo shows how to use `st.write` to visualize Pandas DataFrames. (Data courtesy of the [UN Data Explorer](#).)

Choose countries

United States of ... x China x India x

#### Gross Agricultural Production (\$B)

| Region                   | 1961    | 1962    | 1963    | 1964    | 1965    | 1966    | 1967     | 1968     | 1969     |
|--------------------------|---------|---------|---------|---------|---------|---------|----------|----------|----------|
| China                    | 58.3407 | 60.6909 | 63.9427 | 68.4626 | 74.7479 | 80.4459 | 81.7187  | 81.4748  | 82.1111  |
| India                    | 50.1427 | 50.0158 | 51.3677 | 52.2677 | 49.4983 | 49.3096 | 52.7315  | 55.2082  | 56.1111  |
| United States of America | 89.8166 | 90.275  | 93.7004 | 94.3237 | 97.7041 | 97.2772 | 101.1779 | 103.4206 | 104.1111 |

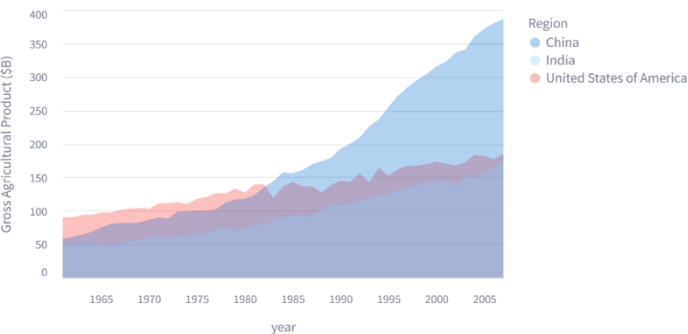


图 3 DataFrame Demo

(2) 可视化地理位置数据，如图 4，在地图上立体化展示了地理数据，地图可以任意地放大缩小和移动，达到实时交互的效果。

### Mapping Demo

This demo shows how to use `st.pydeck_chart` to display geospatial data.

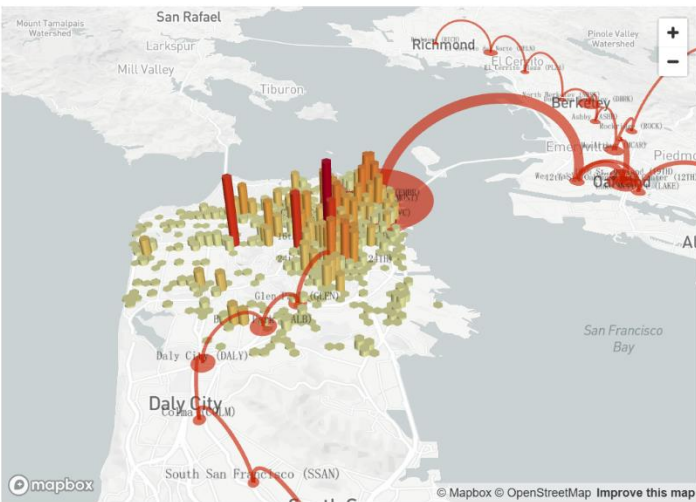


图 4 Mapping Demo

(3) 构建知识图谱，进行实体关系的可视化展示。如下图 5，用 Streamlit 实现了漫威宇宙的建构<sup>4</sup>。

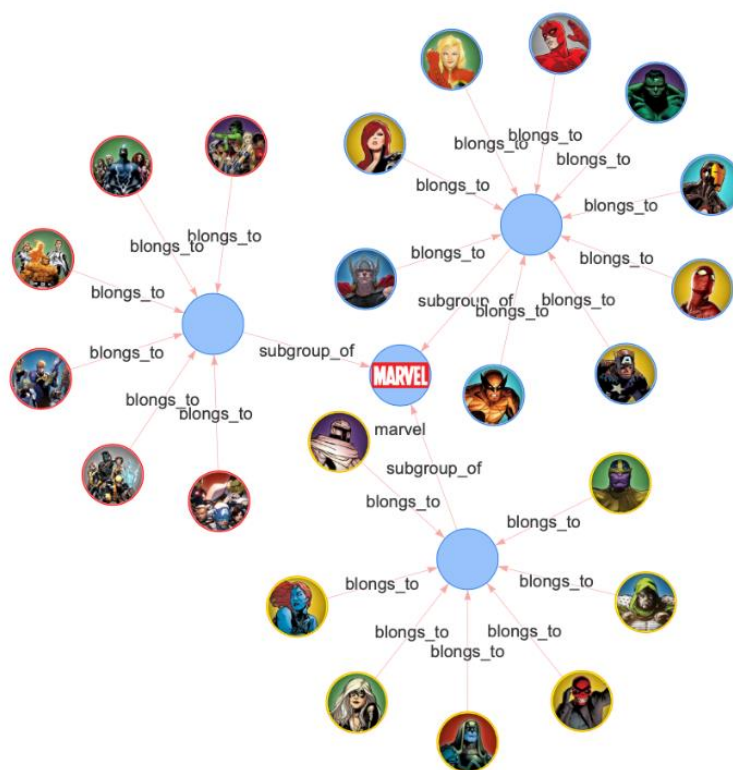


图 5 Marvel

### 3 Streamlit 的常用语法<sup>5</sup>

#### 3.1 页面构建

`st.markdown()` 专门用于渲染 Markdown 格式的文本，可以使用 Markdown 语法来格式化文本，比如标题、列表、粗体、斜体等；`st.title()` 用于创建页面主标题，`st.header()` 创建文章一级标题，`st.subheader()` 创建次级标题，`st.text()` 能输出简单文本，而 `st.write()` 更通用，能输出可以输出各种类型的内容，包括文本、代码块、DataFrame、图像和音频等。

#### 3.2 图表展示

`st.line_chart()` 展示折线图，`st.area_chart()` 展示面积图，`st.bar_chart()`

---

<sup>4</sup> 图片来源：<https://github.com/ChrisDeClea/streamlit-agraph/blob/master/imgs/marvel.png>

<sup>5</sup> 更多详情见 [API Reference - Streamlit Docs](#)

展示条形图，`st.map()`展示地图上的数据点。除此之外，Streamlit 还支持多种图表库，如 Matplotlib、Seaborn、Plotly、Vega-lite 等，以下是几种常见的展示方法：`st.pyplot()`、`st.plotly_chart()`、`st.vega_lite_chart()`。

### 3.3 网页交互

`st.slider()`创建滑块，`st.radio()`创建单选项，`st.checkbox()`创建复选框，`st.selectbox()`创建下拉框，`st.text_input()`让用户能够输入文本，`st.number_input()`让用户能输入数字，`st.file_uploader()`还能支持用户上传文件。

## 4 Streamlit 的安装部署

### 4.1 安装运行

只需在命令行输入 `Pip install streamlit` 即可安装（下载时间较长，需要耐心等待，因为依赖包比较多，如有需要可以先建立虚拟环境，然后在虚拟环境中下载），安装后可以通过运行 `streamlit hello` 指令判断是否安装成功。

运行方法：`import streamlit as st`，在 python 文件中使用 `st` 函数后在终端输入 `Streamlit run xxx.py` (文件名)即可使用运行。

### 4.2 部署分享

Streamlit 的部署方法有很多种，例如 Streamlit Community Cloud、Docker、Heroku、Google Cloud 等，现在着重介绍下 Streamlit Community Cloud。

首先打开 Streamlit Cloud 官网<sup>6</sup>，注册账号管理 GitHub，然后就可以部署自己在 Github 仓库中创建的应用，过程中需要中提前写好配置文件，否则会报 `ModuleNotFoundError`。另外使用 Flask 建构的网站是不能部署在 Streamlit Community Cloud 上的，否则会报 `ValueError`，如图 6 所示。

---

<sup>6</sup> 网址 [Sign in · Streamlit](#)

Traceback:

### 图 6 部署基于 Flask 建构的 app 出现的问题

## 5 总结

## 6 附录

- [一个傻瓜式构建可视化 web 的 Python 神器 -- streamlit - 知乎](https://zhuanlan.zhihu.com/p/100000000)  
(zhihu.com)

<sup>7</sup> 摘自 [Streamlit 简明指南（一） - 知乎 \(zhihu.com\)](#)