

Exploración de la Librería Shiny para Python

Eduardo Valero Vilella

Septiembre 2024

Abstract

Este trabajo presenta una exploración de la librería *Shiny* para *Python*, una librería originalmente del ecosistema de R que ha sido adaptada para Python, que permite el desarrollo de aplicaciones web interactivas centradas en la visualización de datos. Shiny ofrece a los analistas y científicos de datos una herramienta accesible para crear interfaces visuales dinámicas y compartir resultados de manera efectiva.

A través de varios ejemplos prácticos, se estudiarán un conjunto de aplicaciones que exploran funcionalidades clave de Shiny, tales como la reactividad, la integración con bibliotecas de visualización como Matplotlib y Plotly, y su flexibilidad para trabajar con datos dinámicos.

En este documento se describe la estructura de la presentación. De igual forma se usará como guía escrita de la misma.

1 Introducción

1.1 Contexto y Motivación

Explicación de qué rol cumple Shiny en el ecosistema de visualización de datos de Python. También mencionar las principales características de Shiny que la hacen destacar.

1.2 Introducción a Shiny para Python

Explicación de los orígenes de Shiny como librería para *R*, y la empresa que lo desarrolla, *Posit*, también desarrolladora de *RStudio*.

Explicación de la principal funcionalidad de Shiny, la creación de *Dashboards* dinámicos e interactivos en forma de aplicación web, sin necesidad de conocimientos avanzados de desarrollo web.

Comentar algunas alternativas, como *Streamlit* y *Dash*.

1.3 Objetivos del Trabajo

Este trabajo tiene los siguientes objetivos:

- Exploración de Shiny en Python.
- Estudiar la creación de visualizaciones interactivas y dashboards.
- Compararla con otras alternativas.

2 ¿Cómo funciona Shiny?

- Modelo cliente-servidor.
- CLI de Shiny

3 Setup

3.1 Instalación de la librería

```
pip install shiny
```

3.2 Entorno de desarrollo usado

Mencionar que se usan otras librerías, como *pandas*, y que se usa VS Code con el plug-in oficial de Shiny, y con el necesario para editar notebooks de Python.

3.3 Conjuntos de datos utilizados

Breve mención y comentarios a cerca de los datasets utilizados.

4 Ejemplos de uso

4.1 Introducción

Estructura básica de un proyecto de Shiny y la separación entre interfaz de usuario (ui) y la aplicación.

Cómo crear y ejecutar un proyecto usando el Cli de Shiny, y como hacer que los proyectos de Shiny se auto refresquen al editar el código.

```
shiny create
shiny run
shiny run --reload --launch-browser app_dir/app.py
```

4.2 Creación de una Aplicación Simple

Ejemplo de una aplicación muy básica de Shiny para mostrar lo descrito en el apartado anterior en acción para mostrar el funcionamiento más básico, al estilo del slider con un gráfico de Matplotlib que se presenta en la web de Shiny.

4.3 Casos de Uso Avanzados

Aplicación más compleja, como por ejemplo un dashboard interactivo. Introducir gráficos de Plotly para ver como interacciona con Shiny.

5 Conclusiones

5.1 Ventajas y Desventajas

Comentar las ventajas y desventajas que tiene Shiny, comparandolas con otras librerías de Python como Dash y Streamlit.

5.2 Otras conclusiones

Otras conclusiones que se extraigan del trabajo.

6 Bibliografía

- Shiny for Python Documentation: <https://shiny.posit.co/py/docs/>
- Shiny for Python Tutorials: <https://shiny.posit.co/py/tutorials/>
- Comparativa de Shiny, Streamlit: <https://shiny.posit.co/py/docs/comp-streamlit.html>