

Documentación del Código Fuente: Aplicaciones Interactivas de Inferencia Estadística

Y. Alex Rodriguez Velasco

October 28, 2025

Abstract

Este documento proporciona la documentación técnica y conceptual de dos aplicaciones interactivas desarrolladas en Streamlit (Python) para la demostración de principios clave de la Inferencia Estadística: el Teorema de Bayes y la comparación entre enfoques Paramétricos y No Paramétricos.

1 Introducción

El proyecto consta de dos módulos principales desarrollados en Python con el *framework* **Streamlit**. Cada módulo está diseñado para permitir al usuario manipular parámetros clave y observar el impacto estadístico resultante, cumpliendo con los requisitos de la demostración práctica. Las librerías estadísticas clave utilizadas son **numpy**, **scipy** y **plotly** para la visualización dinámica.

2 Módulo 1: Paradoja del Falso Positivo (`bayesian_paradox_app.py`) —*Inferencia Bayesiana*

Este módulo se centra en el **Teorema de Bayes** y la demostración práctica de la **Paradoja del Falso Positivo**. El objetivo es ilustrar cómo la **probabilidad a priori (prevalencia)** impacta desproporcionadamente la **probabilidad a posteriori (VPP)**.

2.1 Fundamento Teórico y Cálculo

El cálculo central se basa en la determinación del **Valor Predictivo Positivo (VPP)**, que es la probabilidad de tener la enfermedad (E) dado un resultado positivo (+):

$$VPP = P(E|+) = \frac{P(+|E) \cdot P(E)}{P(+|E) \cdot P(E) + P(+|E^c) \cdot P(E^c)}$$

- **Variables Clave:** El usuario ajusta interactivamente la Prevalencia ($P(E)$), la Sensibilidad ($P(+|E)$) y la Especificidad ($P(-|E^c)$).
- **Cálculo Dinámico:** La aplicación actualiza el VPP en tiempo real, demostrando que con una Prevalencia baja (ej. 0.1%), el VPP puede ser sorprendentemente bajo ($< 5\%$ para los valores iniciales).

2.2 Componentes de la Interfaz

- **Métricas Clave:** Muestran los valores de entrada y el resultado del VPP con alta precisión.
 - **Visualización:** Un gráfico de pastel (**plotly**) ilustra la paradoja al desglosar el conjunto de resultados positivos en **Verdaderos Positivos** y **Falsos Positivos**, mostrando visualmente la desproporción.
 - **Explicación Matemática:** La sección de información detalla la fórmula de Bayes con los valores dinámicos del cálculo actual, justificando por qué los Falsos Positivos dominan el resultado.
-

3 Módulo 2: Paramétrica vs No Paramétrica (parametric_app.py) —*Pruebas de Hipótesis*

Este módulo contrasta la decisión de utilizar enfoques de inferencia que asumen o no una distribución específica. El caso práctico se centra en la comparación de dos grupos (Control vs. Tratamiento).

3.1 Escenarios Prácticos y Supuestos

La aplicación permite la selección de dos escenarios clave para la demostración:

1. **Escenario Paramétrico (Datos Normales):** Se simulan datos con distribución Normal, cumpliendo los supuestos de la Prueba t.
2. **Escenario No Paramétrico (Datos Asimétricos):** Se simulan datos con distribución Exponencial (sesgada), violando el supuesto de Normalidad.

3.2 Inferencia Aplicada y Justificación

Ambos enfoques se aplican simultáneamente al set de datos para su comparación:

- **Análisis de Supuestos:** Se utiliza la **Prueba de Shapiro-Wilk** y un **Gráfico de Violín** para evaluar el supuesto de Normalidad en los datos seleccionados.
- **Prueba t de Student (Paramétrica):** Compara las medias (μ). Es el método más potente cuando se cumple la Normalidad.
- **Test U de Mann-Whitney (No Paramétrica):** Compara las medianas o rangos. Es el método más fiable cuando se viola la Normalidad.

La justificación final es **dinámica**: para datos normales se recomienda la Prueba t (mayor poder), y para datos asimétricos se recomienda el Test U (mayor robustez), explicando por qué la Prueba t deja de ser válida en este último caso.

4 Requisitos de Ejecución

Para replicar el entorno del proyecto, se requiere Python 3.x y las siguientes librerías, listadas en el archivo `requirements.txt`:

```
streamlit
pandas
numpy
scipy
plotly
```