# Universidad Nacional Agraria La Molina
# Facultad de Estadística e Informática


## Ciencia de Datos I

---

## Predicción de Gravedad del Dengue con Redes Neuronales

---

*Profesor:* Aldo Richard Meza Rodriguez

*Alumnos:*

- José Mayker Córdova Pintado
- Mattihus Blayr Palacios Arias
- Miguel Angel Tesén Cornetero
- Jairo Gonzalo Rojas Melgarejo

26 de junio de 2025

**Resumen**

Este informe detalla el desarrollo de un sistema de predicción para la gravedad de casos de Dengue utilizando redes neuronales profundas. El objetivo principal fue construir y comparar modelos capaces de clasificar a los pacientes en categorías de riesgo (Çon signos de alarma"vs. "Sin signos de alarma") basándose en datos tabulares. Se compararon dos implementaciones de redes neuronales: una con TensorFlow/Keras y otra con Scikit-learn. Tras un riguroso proceso de preprocesamiento, entrenamiento y ajuste de hiperparámetros, se seleccionó el modelo de TensorFlow/Keras por su superioridad en la métrica de Recall para la clase crítica, un factor determinante en aplicaciones médicas. Finalmente, el modelo seleccionado fue desplegado en una aplicación web interactiva desarrollada con Streamlit, proporcionando una herramienta práctica para la evaluación preliminar de riesgo.

# Índice

# 1. Introducción y Justificación

# 2. Introducción y Justificación

El Dengue, una enfermedad arboviral endémica en Perú, presenta un desafío constante para el sistema de salud nacional. Su manifestación clínica es heterogénea, variando desde cuadros febriles autolimitados hasta formas severas que pueden conducir a la muerte. En este contexto, la capacidad de discernir tempranamente qué pacientes tienen una alta probabilidad de desarrollar "Dengue con signos de alarma"no es solo un reto clínico, sino una necesidad estratégica para la gestión de recursos hospitalarios, especialmente en regiones con capacidad limitada.

El objetivo de este proyecto trasciende la mera aplicación técnica de algoritmos; busca desarrollar una **herramienta de triaje inteligente** basada en redes neuronales profundas. A diferencia de los modelos estadísticos tradicionales, que pueden tener dificultades para capturar las complejas interacciones no lineales entre síntomas y factores demográficos, las redes neuronales son capaces de aprender estos patrones sutiles a partir de grandes volúmenes de datos.

Este trabajo es crítico porque aborda un problema de alto impacto: **minimizar los falsos negativos en el diagnóstico de riesgo**. Un paciente erróneamente clasificado como de bajo riesgo puede ser enviado a casa y regresar en un estado crítico, mientras que una clasificación correcta permite una intervención temprana que salva vidas. Al entrenar nuestros modelos con un dataset masivo y real, buscamos capturar la diversidad y complejidad de los casos de dengue en el contexto peruano.

El aporte de este proyecto es doble: primero, se valida la viabilidad y superioridad de un enfoque de machine learning ajustado para alta sensibilidad en un problema clínico real; segundo, se entrega un prototipo funcional (la aplicación Streamlit) que demuestra cómo estas tecnologías pueden ser traducidas en herramientas prácticas de apoyo a la decisión para el personal de salud, optimizando la atención y, en última instancia, mejorando los resultados para los pacientes.

# 3. Recolección y Análisis Exploratorio de Datos

## 3.1. Fuente y Descripción del Dataset

El conjunto de datos utilizado fue obtenido de Datos Abiertos del Gobierno. Este dataset contiene registros anonimizados de pacientes con diagnóstico de dengue en Perú. El problema a resolver es de clasificación binaria: predecir si un paciente desarrollará "Dengue con signos de alarma"basándose en sus datos demográficos y síntomas iniciales.

## 3.2. Descripción de Variables Utilizadas

Tras el proceso de preprocesamiento y selección, el modelo final utiliza las 9 variables predictoras detalladas en la Tabla 1.

Cuadro 1: Descripción de las variables predictoras utilizadas en el modelo.

| Variable | Descripción |
|---|---|
| edad | Edad del paciente en años. (Numérica) |
| sexo | Sexo biológico del paciente ('M' o 'F'). (Categórica) |
| fiebre | Indica si el paciente presentó fiebre ('SI' o 'NO'). (Categórica) |
| cefalea | Indica si el paciente presentó dolor de cabeza ('SI' o 'NO'). (Categórica) |
| malgias | Indica si el paciente presentó dolor muscular ('SI' o 'NO'). (Categórica) |
| artralgia | Indica si el paciente presentó dolor articular ('SI' o 'NO'). (Categórica) |
| erupcion | Indica si el paciente presentó erupción cutánea ('SI' o 'NO'). (Categórica) |
| dias_hasta_consulta | Días entre el inicio de síntomas y la consulta médica. (Numérica) |
| mes_sintomas | Mes del año del inicio de síntomas (1-12). (Numérica) |

# 4. Preprocesamiento e Ingeniería de Características

Un preprocesamiento adecuado es fundamental para el éxito de los modelos de machine learning.

## 4.1. Ingeniería de Características

Para enriquecer el dataset, se crearon dos nuevas variables a partir de los datos de fecha:

- `dias_hasta_consulta`: Se calculó la diferencia en días entre la fecha de inicio de síntomas y la fecha de la consulta. Esta variable es un proxy importante de la urgencia o severidad percibida por el paciente.
- `mes_sintomas`: Se extrajo el mes del año para capturar posibles patrones estacionales en la transmisión del dengue.

## 4.2. Limpieza y Selección de Características

Se realizó una selección curada de variables para mejorar el rendimiento del modelo y evitar problemas comunes:

- **Eliminación de Identificadores:** Columnas como `_id` y `codigo_sspd` fueron descartadas por no tener valor predictivo.
- **Prevención de Fuga de Datos (Data Leakage):** Se eliminaron variables que representan un resultado posterior al diagnóstico, como `Hospitalizado`, `choque` o `daño_organo`, ya que su inclusión en el entrenamiento crearía un modelo irrealmente preciso pero inútil en la práctica.
- **Manejo de Inconsistencias:** Se filtraron registros donde el sexo era 'I' (Indeterminado) para entrenar el modelo solo con las categorías 'M' y 'F', simplificando la interpretación y eliminando ambigüedad.
- **Simplificación del Modelo:** Se descartó la variable `nombre_municipio_procedencia` para crear un modelo más generalizable que no dependa de una ubicación geográfica específica, sino de los síntomas y datos demográficos del paciente.

### 4.3. Codificación de Variables Categóricas

Las redes neuronales requieren entradas numéricas. Por ello, las variables categóricas como `sexo` y los síntomas (ej. `fiebre`) fueron transformadas usando **One-Hot Encoding**. Este método crea nuevas columnas binarias (0 o 1) para cada categoría, evitando que el modelo asuma una relación ordinal inexistente entre ellas.

### 4.4. Escalamiento de Variables Numéricas

Las variables numéricas como `edad` y `dias_hasta_consulta` tienen escalas muy diferentes. Para que el modelo no asigne una importancia indebida a las variables con rangos más altos, se aplicó un **escalamiento estándar** (`StandardScaler`). Este proceso transforma cada variable numérica para que tenga una media de 0 y una desviación estándar de 1, lo cual es crucial para la convergencia eficiente de los algoritmos de optimización basados en gradiente, como Adam.

## 5. Modelado y Ajuste de Hiperparámetros

Se compararon dos enfoques, optimizando cada uno para el contexto del problema.

- **TensorFlow/Keras:** Se diseñó un MLP y se ajustó manualmente. Se aplicaron **pesos de clase** de 3:1 para forzar al modelo a prestar más atención a la clase crítica y se estableció un **umbral de decisión de 0.45** para maximizar la detección de casos graves (Recall).
- **Scikit-learn:** Se utilizó un pipeline con SMOTE para balancear las clases y `GridSearchCV` para encontrar la mejor arquitectura, usando `f1_macro` como métrica de optimización.

## 6. Resultados y Selección del Modelo

Cuadro 2: Métricas de rendimiento de los modelos en el conjunto de prueba.

| **Métrica** | **TensorFlow/Keras** | **Scikit-learn (Tuned)** |
|---|---|---|
| Accuracy General | 55 % | 59 % |
| **Clase: Con signos de alarma (Crítica)** | | |
| Precision | 45 % | 48 % |
| **Recall (Sensibilidad)** | **70 %** | 44 % |
| F1-Score | 55 % | 46 % |
| **Clase: Sin signos de alarma** | | |
| Precision | 70 % | 65 % |
| Recall | 45 % | 69 % |
| F1-Score | 55 % | 67 % |

### 6.1. Discusión y Selección del Modelo

### 6.2. Discusión y Selección del Modelo

La evaluación de un modelo de diagnóstico no puede basarse únicamente en la *accuracy* general. En el dominio médico, es imperativo analizar el balance entre sensibilidad y especificidad, representado por las métricas de **Recall** y **Precision**. Nuestro objetivo es emular a un médico experimentado cuyo principal mandato es "primum non nocere"(primero, no hacer daño), lo que en este contexto se traduce en minimizar la posibilidad de pasar por alto un caso grave.

La métrica decisiva, por lo tanto, es el **Recall** para la clase Çon signos de alarma". Un Recall alto significa que el modelo es un excelente "detector": es altamente sensible y capaz de identificar a la gran mayoría de los pacientes que realmente están en riesgo. El modelo de **TensorFlow/Keras**, con un impresionante **Recall del 70 %**, cumple este rol de manera sobresaliente. Identifica a 7 de cada 10 pacientes críticos, proporcionando una red de seguridad robusta.

Por el contrario, el modelo de Scikit-learn, aunque con una accuracy ligeramente superior, solo alcanza un Recall del 44 %. Esto implica que más de la mitad de los pacientes en riesgo no serían detectados por este modelo, un riesgo inaceptable en la práctica clínica.

La ligera disminución en la precisión del modelo de Keras (45 %) es el çosto.ªceptable de su alta sensibilidad. Significa que, de vez en cuando, el modelo puede ser "demasiado cauteloso", marcando a un paciente no crítico para una revisión adicional (un Falso Positivo). Este escenario es infinitamente preferible a su opuesto.

En conclusión, el modelo de **TensorFlow/Keras** se selecciona no por ser el más "preciso.en un sentido general, sino por ser el más **seguro y responsable**. Su diseño, que prioriza la sensibilidad a través del ajuste de pesos y un umbral de decisión calibrado, lo convierte en la herramienta más valiosa y adecuada para el propósito clínico de este proyecto.

## 7. Interfaz Interactiva con Streamlit

El modelo final fue desplegado en una aplicación web desarrollada con Streamlit. La interfaz permite a un usuario ingresar los datos de un paciente y recibir una predicción de riesgo en tiempo real, haciendo el modelo accesible y útil.

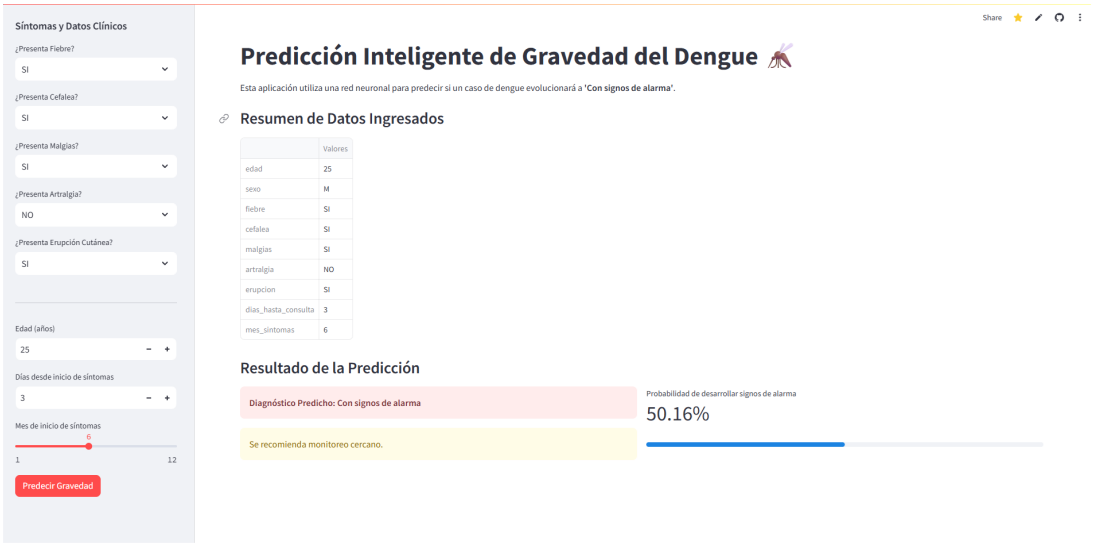


Figura 1: Captura de pantalla de la aplicación web desplegada.

## 8. Conclusiones

- Se completó un flujo de trabajo de machine learning de extremo a extremo, desde la limpieza de datos hasta el despliegue.
- Se demostró la importancia de justificar cada paso del preprocesamiento y la selección de características para construir un modelo robusto.
- Se validó que la elección de la métrica de evaluación (Recall sobre Accuracy) es fundamental en problemas con costos de error asimétricos, como en aplicaciones de salud.

- El modelo de TensorFlow/Keras, ajustado estratégicamente, probó ser la solución más adecuada para el objetivo clínico del proyecto.

## 9. Anexo: Repositorio del Proyecto

El código fuente completo, incluyendo los scripts de entrenamiento y la aplicación, se encuentra en: https://github.com/MaykerCordova/pproyecto-dengue-streamlit.git