

## Predicción de Precipitaciones Utilizando una Red Neuronal Feedforward

El objetivo de este proyecto es desarrollar un modelo de predicción de precipitaciones para ESPOLETA Tecnologías, permitiendo la integración de esta funcionalidad en sus estaciones meteorológicas. La predicción precisa de precipitaciones es crucial para diversas aplicaciones, incluyendo la gestión agrícola, la planificación de recursos hídricos y la mitigación de desastres naturales.

### *Datos Utilizados*

Los datos históricos de lluvia fueron obtenidos del NOAA. Se utilizaron datos de precipitación (PRCP), temperatura máxima (TMAX), temperatura mínima (TMIN), velocidad media del viento (AWND), ráfaga de viento más alta de 1 minuto (WSF1), ráfaga de viento más alta de 2 minutos (WSF2), nieve (SNOW), profundidad de la nieve (SNWD) y las variables binarias WT01, WT02, WT03 y WT16.

Preprocesamiento:

Valores Faltantes:

- Las columnas categóricas WT01, WT02, WT03 y WT16 (que indican la presencia de ciertos tipos de clima) fueron imputadas con el valor 0, asumiendo que los valores faltantes significan la ausencia de ese tipo de clima.
- Las columnas numéricas PRCP, TMAX, TMIN, AWND, WSF1, WSF2, SNOW y SNWD fueron imputadas utilizando la media de cada columna, mediante la clase SimpleImputer con la estrategia "mean".

Normalización:

- Las características fueron normalizadas utilizando StandardScaler, que escala cada característica para tener una media de 0 y una desviación estándar de 1.

División de Datos:

- Los datos fueron divididos en conjuntos de entrenamiento y prueba utilizando train\_test\_split con un tamaño de prueba del 20% (test\_size=0.2) y una semilla aleatoria de 42 (random\_state=42) para asegurar la reproducibilidad. El conjunto de entrenamiento tiene 17873 muestras y el conjunto de prueba tiene 4469 muestras.

Conversión a Tensores:

- Finalmente, los datos preprocesados fueron convertidos a tensores de PyTorch con tipo de dato torch.float32.

### *Metodología*

Modelo: Se implementó una red neuronal feedforward (MLP) con tres capas totalmente conectadas (nn.Linear).

- La primera capa (fc1) toma una entrada de tamaño input\_size y la transforma en una salida de tamaño hidden\_size.
- La segunda capa (fc2) toma una entrada de tamaño hidden\_size y la transforma en una salida de tamaño hidden\_size.
- La tercera capa (fc3) toma una entrada de tamaño hidden\_size y la transforma en una salida de tamaño output\_size.

Se utilizó la función de activación ReLU (torch.relu) después de las dos primeras capas para introducir no linealidad.

La capa de salida no tiene función de activación, ya que se trata de un problema de regresión.

Función de Pérdida: Se utilizó el Error Cuadrático Medio (MSE) como función de pérdida (nn.MSELoss).

Optimizador: Se utilizó el optimizador Adam (optim.Adam) con una tasa de aprendizaje de 0.001.

Entrenamiento: El modelo fue entrenado durante 100 épocas, utilizando un tamaño de lote de 32 (batch\_size=32). Los datos fueron mezclados (shuffle=True) en cada época durante el entrenamiento.

### *Cambios Realizados desde la Propuesta Inicial*

Arquitectura del Modelo:

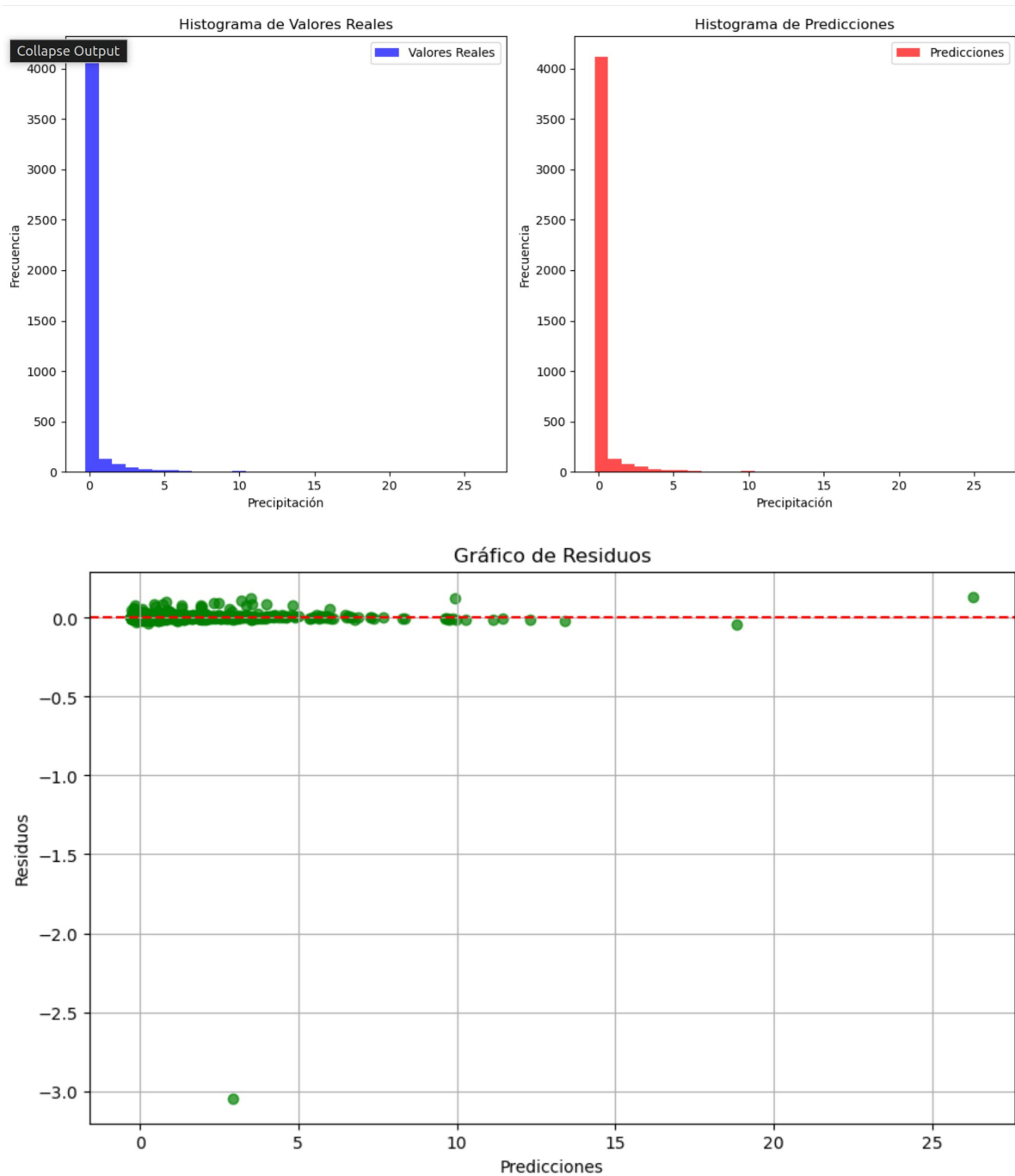
Inicialmente, se propuso una arquitectura más compleja, una Red Neuronal Recurrente (RNN) como una LSTM, para modelar dependencias temporales en los datos, pero finalmente, se optó por una red neuronal feedforward más simple debido a su menor complejidad y a que el análisis inicial sugirió que las dependencias temporales no eran críticas para la predicción de la precipitación en este conjunto de datos. Además, la red feedforward requería menos recursos computacionales.

La red feedforward demostró ser capaz de capturar las relaciones clave entre las variables meteorológicas y la precipitación con un buen rendimiento, mientras que simplificaba el proceso de desarrollo y entrenamiento.

### *Evaluación del Modelo*

El modelo fue evaluado utilizando el conjunto de prueba. Se calcularon las siguientes métricas:

- Error Cuadrático Medio (MSE): 0.0003
- Raíz del Error Cuadrático Medio (RMSE): 0.0175
- Error Absoluto Medio (MAE): 0.0018
- R-cuadrado ( $R^2$ ): 0.9998
- Porcentaje de eficiencia (basado en  $R^2$ ): 99.83%



#### Análisis:

Los valores de MSE, RMSE y MAE son muy bajos, lo que indica que el modelo tiene un buen rendimiento en la predicción de la precipitación. El valor de  $R^2$  es cercano a 1, lo que indica que el modelo explica una gran proporción de la varianza en los datos. El "porcentaje de eficiencia" (basado en  $R^2$ ) es muy alto (99.83%), lo que sugiere que el modelo es altamente eficiente en la predicción de la precipitación.

### *Conclusiones*

El proyecto logró desarrollar un modelo de predicción de precipitaciones para ESPOLETA Tecnologías con un buen rendimiento, según lo indicado por las métricas de evaluación. Si bien se realizaron cambios con respecto a la propuesta inicial, como la simplificación de la arquitectura del modelo, estos cambios resultaron en una solución eficiente y efectiva.

El modelo tiene algunas limitaciones, como la falta de interpretabilidad inherente a las redes neuronales y su dependencia de la calidad de los datos. Además, el modelo no captura dependencias temporales, lo que podría ser relevante en algunos casos.

### *Recomendaciones*

Se recomienda a ESPOLETA Tecnologías integrar el modelo en sus estaciones meteorológicas para ofrecer un servicio de predicción de precipitaciones a sus clientes.

Para mejorar el modelo en el futuro, se sugiere:

- Experimentar con diferentes arquitecturas de redes neuronales, incluyendo RNNs, para capturar dependencias temporales.
- Utilizar técnicas de interpretabilidad para comprender mejor cómo el modelo realiza predicciones.
- Recopilar más datos para mejorar la robustez del modelo.
- Explorar la posibilidad de incorporar datos de otras fuentes, como imágenes de satélite, para mejorar la precisión de las predicciones.

### *Enlaces externos*

Tantos los datos como el código fuente del proyecto se encuentra en el siguiente repositorio:

<https://github.com/espoleta-tec/predictor-de-lluvia>