Universidad Católica del Maule
Ingeniería en Estadística

Clasificación y pronóstico de precipitaciones en Rapa Nui usando XGBoost y técnicas de minería de datos sobre registros meteorológicos.

Rossemari Gajardo González & Angel Llanos Herrera

Profesor: Rodrigo Rivera

Data Mining
IES 411
Ingeniería en Estadística

June 28, 2025

Contents

Chapter 1

Resumen

La Isla de Pascua (Rapa Nui) depende casi exclusivamente de la captación de aguas lluvias para consumo y agricultura, enfrentando precipitaciones irregulares que dificultan la planificación hídrica y agrícola. Este estudio propone un enfoque en minería de datos para modelar y predecir la ocurrencia y cantidad de lluvia en intervalos horarios, con el fin de optimizar el uso del recurso hídrico local. Se utilizó un conjunto de 54.894 registros horarios de la estación Mataveri (código 270001) obtenidos de la Dirección Meteorológica de Chile, que incluyen variables climáticas (temperatura del aire/punto de rocío, humedad, presión, condición de visibilidad del cielo, entre otras.), al igual que precipitaciones por minuto.

Tras agrupar los datos de minutos a hora y limpiar los valores faltantes (imputados mediante KNN Imputer con selección de k basada en RMSE mínimo), se generó la variable binaria lluvia (umbral mayor a 0.4 mm/h), y se normalizaron escalas numéricas con MinMaxScaler. Por otra parte, las variables categóricas fueron codificadas como variables dummies.

El análisis exploratorio confirmó baja frecuencia de agua caída, con una ligera estacionalidad y correlaciones leves (y significativas) con humedad, punto de rocío y presión.

Para los modelos se empleó XGBoost:

1. Clasificación de lluvia vs. no lluvia (binary:logistic) con validación cruzada a 5 folds y una búsqueda aleatoria de hiperparámetros. El mejor alcanzó un balanced_accuracy de 0.854 con un 89% de detección de lluvia y 81% de detección de no lluvia, aunque presentando elevadas tasas de falsos positivos. Sin embargo, a pesar de no conseguir una buena clasificación para la lluvia, este modelo, incluso sin incluir mejoras, puede llevar a la reducción de perdidas por sequía (viendo clasificaciones de no lluvia). La reducción de los falsos positivos (clasificar lluvia cuando realmente no la hubo) mejoraría drásticamente el modelo.

2. Regresión de volumen de precipitación (reg:squarederror), optimizada por RMSE negativo en la búsqueda de parámetros, también con una validación cruzada de 5 folds. Obteniendo un RMSE de 0.8903 mm/h y un coeficiente de determinación R^2 de 0.1161 en el conjunto de validación, indicando una clara subestimación de eventos extremos, explicando mínimamente la variabilidad en las precipitaciones reales. No obstante, capturando momentos de lluvia en la mayoría de ocasiones, pero no correctamente su magnitud.

Se concluye que, si bien XGBoost capta eficazmente la ocurrencia de lluvias en un entorno desbalanceado, su capacidad de cuantificar magnitudes es limitada por la falta de variables que expliquen la mayor variabilidad y momentos de lluvias extremas (sugiriendo la integración de variables como inestabilidades o anomalías climáticas). Se recomienda enriquecer el conjunto de variables con datos satelitales como densidad de nubes, anomalías, entre otros. Además de la definición de modelos estacionales.

Chapter 2

Introducción

La Isla de Pascua (Rapa Nui) es uno de los territorios más remotos del planeta, enfrenta serias limitaciones en el acceso a fuentes de agua dulce. Debido a su aislamiento geográfico, la ausencia de ríos permanentes y su clima subtropical con precipitaciones irregulares, la isla depende casi exclusivamente de la captación de aguas lluvias para el consumo humano y la agricultura. Esta dependencia, sumada a la creciente presión sobre los recursos naturales, hace que la planificación hídrica y agrícola deba considerar enfoques más eficientes y predictivos. En este contexto, el uso de herramientas de ciencia de datos se presenta como una alternativa viable para abordar la incertidumbre climática y optimizar el uso del agua disponible.

El propósito principal de este estudio es explorar y modelar la ocurrencia de precipitaciones en Rapa Nui a partir de variables meteorológicas históricas, utilizando técnicas de minería de datos. Esto se concreta en los siguientes objetivos específicos:

- Visualizar patrones climáticos presentes en los registros meteorológicos de la isla.
- Identificar las variables que más inciden en la ocurrencia y cantidad de lluvia.
- Desarrollar un modelo predictivo que permita anticipar eventos de lluvia con base en condiciones atmosféricas observadas.
- Proveer información que contribuya a la toma de decisiones en la planificación agrícola y la gestión hídrica local.

Este estudio se enfoca en la predicción de precipitaciones en Rapa Nui mediante el uso de técnicas de minería de datos aplicadas a datos meteorológicos horarios registrados por la estación Mataveri. El análisis contempla la exploración, transformación y modelamiento de variables climáticas para identificar patrones asociados a la ocurrencia de lluvia. Si bien el enfoque es predictivo, también se considera el análisis descriptivo de los datos como una etapa esencial para la comprensión del fenómeno climático en la isla.

El alcance de este trabajo se enfoca en la modelación de la variable lluvia como variable binaria (ocurre o no ocurre), a partir de condiciones atmosféricas observadas previamente. No se busca predecir la cantidad exacta de precipitación, sino generar una herramienta que permita anticipar la ocurrencia de eventos lluviosos con fines de apoyo a la planificación local.

La justificación radica en la necesidad urgente de optimizar el uso del agua en un territorio con recursos hídricos extremadamente limitados. En este sentido, la implementación de soluciones basadas en ciencia de datos representa una alternativa sustentable y escalable para fortalecer la resiliencia agrícola de la isla, frente a los desafíos que impone el cambio climático y la incertidumbre meteorológica.

2.1 Planteamiento del problema: Predicción de lluvias en Rapa Nui

La variabilidad e irregularidad de las precipitaciones en Rapa Nui impiden establecer patrones confiables para la planificación de actividades agrícolas. En consecuencia, se produce un uso ineficiente del agua de lluvia, que es el principal recurso hídrico de la isla. Dado que no se cuenta con sistemas hídricos permanentes ni reservas artificiales de gran escala, la falta de información predictiva representa una amenaza para la seguridad alimentaria y la sostenibilidad de los sistemas agrícolas locales.

La presente investigación propone abordar esta problemática mediante el desarrollo de un modelo predictivo que permita anticipar la ocurrencia de lluvia, utilizando datos meteorológicos observados. Esta herramienta busca entregar apoyo a la toma de decisiones relacionadas con el riego, la siembra y la cosecha, incorporando técnicas de minería de datos como un recurso estratégico frente al cambio climático y la escasez hídrica.

2.2 Descripción de los datos

Los datos meteorológicos utilizados en este proyecto fueron obtenidos desde la Dirección Meteorológica de Chile (DMC), una institución encargada del monitoreo, análisis y difusión de información climática a nivel nacional. Para este estudio, se accedió a su sistema público de descarga de datos proveniente de estaciones meteorológicas automáticas, disponible a través de la página web `https://climatologia.meteochile.gob.cl`. Esta plataforma permite consultar y descargar registros horarios históricos de múltiples variables climáticas medidas en distintas estaciones del país. En particular, se seleccionaron los registros correspondientes a la estación "Mataveri Isla de Pascua Ap.", ubicada en Rapa Nui, la cual se encuentra identificada bajo el código nacional 270001.

El conjunto de datos contiene un total de 54.894 observaciones, correspondientes a mediciones horarias recopiladas en un período continuo. Se dispone de 12 variables, incluyendo tanto características atmosféricas como marcadores temporales y categóricos.

A continuación, se describen las variables en el conjunto de datos utilizado en este estudio, junto con su significado y unidad de medida correspondiente:

- **codigoNacional**: Código numérico único de la estación meteorológica.
- **momento**: Fecha y hora (UTC) de cada registro.
- **rr1Hora**: Precipitación acumulada (mm) en la última hora.
- **ts**: Temperatura del aire seco (°C).
- **td**: Temperatura del punto de rocío (°C).
- **hr**: Humedad relativa del aire (%).
- **qff**: Presión atmosférica reducida al nivel del mar (hPa).
- **ddInst**: Dirección del viento (°) en el instante de medición.
- **ffInst**: Intensidad del viento en el instante de medición.
- **isSkyClear**: Condición de visibilidad del cielo.
- **mes**: Mes en el que se registró la observación.
- **estacion**: Estación del año correspondiente al mes del registro.

Datos y acceso a los códigos utilizados tanto en R como en Python para llevar a cabo el proyecto en el siguiente repositorio en GitHub.

https://github.com/angellds3/XGBoost_Regresion_Clasificacion_Lluvia_Rapa_Nui

Chapter 3

Preparación y comprensión de los datos

3.1 Limpieza y transformación

Para llevar a cabo el análisis, se realizó un proceso de limpieza y transformación de los datos, comenzando con una exploración inicial que permitió identificar el tipo de variables disponibles. La base de datos original contenía ocho variables numéricas continuas, una variable numérica entera y dos variables tipo objeto, que fueron correctamente reinterpretadas como variables categóricas.

Durante esta etapa, también se detectaron valores faltantes en distintas variables: doce valores en 'rr1Hora' (precipitación acumulada en la última hora) y 'isSkyClear' (condición de cielo visible), nueve valores en 'qff' (presión atmosférica a nivel del mar en hPa), y un valor faltante en cada una de las variables 'ddInst' y 'ffInst' (dirección e intensidad del viento al instante de medición, respectivamente).

Posteriormente, se realizó una transformación adecuada de las variables categóricas, definiendo como tales a 'isSkyClear', 'mes' y 'estacion'. Asimismo, para facilitar la interpretación y el análisis posterior, se estandarizó la nomenclatura de las variables de acuerdo con los siguientes criterios:

- **codigoNacional**: codigo_estacion
- **momento**: momento_registro
- **rr1Hora**: agua_caida
- **ts**: temperatura_aire
- **td**: temperatura_puntorocio
- **hr**: humedad_aire
- **qff**: presion
- **ddInst**: direccion_viento
- **ffInst**: intensidad_viento
- **isSkyClear**: cielo_visible
- **mes**: mes
- **estacion**: estacion

De igual forma, se eliminó la variable 'codigo_estacion' por corresponder a un identificador único y constante que no aporta variabilidad ni contenido informativo útil para tareas de predicción. Esta depuración permitió trabajar con un conjunto de variables más limpio y enfocado. También se generaron dos nuevas variables relevantes para el análisis:

- **lluvia**: Variable binaria definida como 1 si el valor de 'agua_caida' es mayor a 0.4 mm, y 0 en caso contrario.
- **periodo**: Categoría temporal según el momento del día en que se registró la medición, definida de la siguiente manera:

- Madrugada: 00:00–05:59
- Mañana: 06:00–11:59
- Tarde: 12:00–17:59
- Noche: 18:00–23:59

A partir de aquí, se aplicaron las transformaciones requeridas para el análisis posterior con datos limpios e imputados. En primer lugar, se eliminaron las variables 'direccion_viento' e 'intensidad_viento', debido a que su correlación con la variable de interés fue menor a 0.05. Además, ambas contenían un único valor faltante, por lo que se optó por excluirlas del análisis.

Para el tratamiento de los valores faltantes restantes, se utilizó el método del vecino más cercano (KNN Imputer). Este método interpreta cada observación como un punto en un espacio multidimensional y estima los valores ausentes a partir de los registros más cercanos. El espacio multidimensional definido para optimizar resultados y conseguir una mayor representatividad según correlaciones con las variables con valores faltantes a estimar son: Para agua_caida se utilizó el espacio de su propia variable, temperatura_puntorocio, humedad_aire y presion. Luego para presion se utilizó el espacio de su propia variable y temperatura_puntorocio. Finalmente, para cielo visible se utilizó el espacio de su propia variable, agua_caida, humedad_aire, temperatura_puntorocio. Por otra parte, la cantidad óptima de vecinos (k) se definió a partir del cálculo del error cuadrático medio (RMSE) usando el 20% de los datos con valores completos. Entregando los siguientes resultados.

Table 3.1: Imputación de valores faltantes por KNN para distintos valores de `k-vecinos`

Variable	k=3	k=5	k=7	k=10	k=15
agua_caida	0.7268	0.6955	0.6869	0.6715	**0.6574***
cielo_visible	0.6502	0.6426	0.6307	**0.6116***	0.6149
presion	4.4491	4.2162	4.1115	4.0303	**3.9705***

Se seleccionaron los siguientes k:

- **agua_caida**: $k = 15$
- **cielo_visible**: $k = 10$
- **presion**: $k = 15$

Al aplicar esta imputación se generaron las siguientes estimaciones sobre los valores faltantes.

Table 3.2: Valores faltantes estimados por KNN (en * los valores imputados)

Momento	**Agua caída**	**Temp. aire**	**Temp. punto rocío**	**Humedad**	**Presión**	**Cielo vis.**	**Mes**	**Estación**	**Periodo**
2019-10-11 22:00	0.000	21.153	14.480	65.200	1018.387	0.0*	Octubre	Primavera	Noche
2019-10-12 00:00	0.000	19.522	13.817	69.278	1019.250	0.0*	Octubre	Primavera	Madrugada
2019-11-27 01:00	0.000	20.860	16.058	73.650	1022.395	0.0*	Noviembre	Primavera	Madrugada
2023-03-03 04:00	0.000	23.300	19.500	79.000	1019.889*	0.0	Marzo	Verano	Madrugada
2024-11-01 07:00	0.000	17.048	10.072	63.200	1025.550	0.0*	Noviembre	Primavera	Mañana
2019-10-01 01:00	0.0133*	19.427	15.963	80.183	1022.378	1.0	Octubre	Primavera	Madrugada
2019-10-01 02:00	0.0133*	19.560	16.440	81.900	1023.000	0.0	Octubre	Primavera	Madrugada
2019-10-01 03:00	0.000*	19.747	16.900	83.383	1023.497	0.0	Octubre	Primavera	Madrugada
2019-10-01 04:00	0.0533*	19.402	16.992	85.867	1023.717	1.0	Octubre	Primavera	Madrugada
2024-11-01 08:00	0.000*	16.973	10.428	65.050	1023.186*	0.0*	Noviembre	Primavera	Mañana
2024-11-01 09:00	0.000*	17.157	10.295	63.750	1025.675*	0.0*	Noviembre	Primavera	Mañana
2024-11-01 10:00	0.000*	17.065	10.403	64.583	1023.675*	0.0*	Noviembre	Primavera	Mañana
2024-11-01 11:00	0.000*	16.787	10.105	64.383	1022.398*	0.0*	Noviembre	Primavera	Mañana
2024-11-01 12:00	0.000*	17.522	9.833	60.350	1025.083*	0.0*	Noviembre	Primavera	Tarde
2024-11-01 13:00	0.000*	17.850	9.888	59.233	1023.664*	0.0*	Noviembre	Primavera	Tarde
2024-11-01 14:00	0.000*	18.494	10.094	57.731	1024.321*	0.0*	Noviembre	Primavera	Tarde
2024-11-01 15:00	0.000*	19.137	10.098	55.327	1023.871*	0.0*	Noviembre	Primavera	Tarde

Luego, se aplicaron las transformaciones finales para preparar los datos para el modelo XGBoost:

- Se normalizaron las variables numéricas 'temperatura_aire', 'temperatura_puntorocio', 'humedad_aire' y 'presion' utilizando la función MinMaxScaler.
- Se codificaron las variables categóricas 'cielo_visible', 'mes' y 'periodo' usando codificación one-hot ('get_dummies').
- El conjunto de datos resultante fue almacenado en 'df_final', el cual representa la base imputada, normalizada y codificada, lista para el análisis y modelado.

3.2 Análisis exploratorio de los datos

Antes de aplicar las transformaciones, se realizó un análisis univariado que permitió caracterizar las variables meteorológicas originales. La variable 'agua_caida', que representa la precipitación acumulada por hora, presentó una distribución altamente asimétrica hacia valores bajos. La mediana fue cero y la media fue de solo 0.106 mm/h, lo cual refleja una alta frecuencia de horas sin lluvia. Sin embargo, se observaron valores máximos de hasta 33.4 mm/h, lo que evidencia la presencia de eventos extremos de precipitación, poco frecuentes.

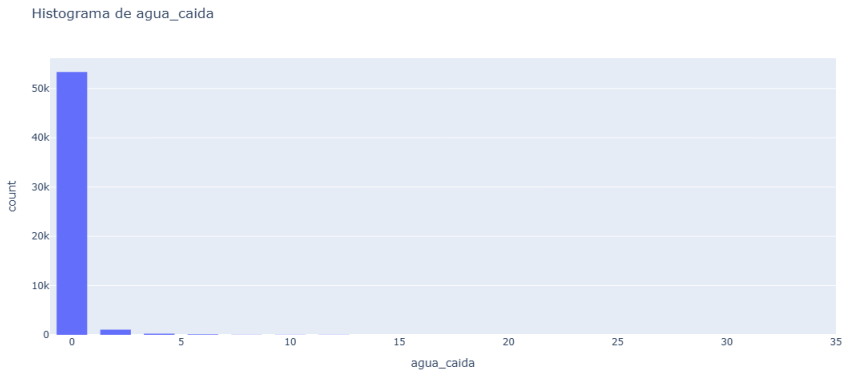


Table 3.3: Estadísticas descriptivas de variables meteorológicas

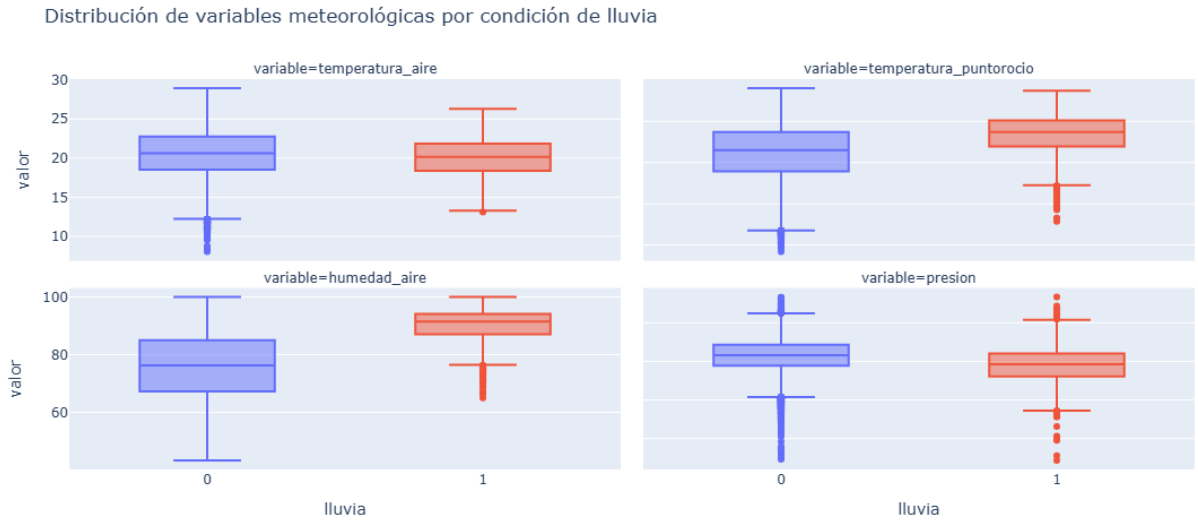
Estadístico	agua_caida	temperatura_aire	temperatura_puntorocio	humedad	presion	cielo_visible
count	54882	54894	54894	54894	54885	54882
mean	0.106	20.625	16.231	76.481	1021.408	0.373
min	0.000	8.047	4.1967	43.300	994.326	0.000
25%	0.000	18.528	14.043	67.633	1018.757	0.000
50%	0.000	20.616	16.618	76.967	1021.477	0.00
75%	0.000	22.692	18.808	85.867	1024.240	1.000
max	33.400	28.947	24.035	100.000	1036.660	1.000
std	0.723	3.045	3.295	11.386	4.334	0.484

Las variables térmicas 'temperatura_aire' y 'temperatura_puntorocio' mostraron distribuciones más simétricas y estables, con medias de 20.6 °C y 16.2 °C, respectivamente. Estas variables reflejan las condiciones climáticas templadas de Rapa Nui y su diferencia es un indicador indirecto de la saturación del aire, lo cual es relevante para la condensación y formación de lluvias. La presión atmosférica ('presion') presentó baja variabilidad, con una media cercana a 1021 hPa, lo que sugiere condiciones atmosféricas mayormente estables. En contraste, las variables de viento presentaron una dispersión más amplia, donde la dirección osciló con alta variabilidad en torno a 155°, mientras que la intensidad del viento alcanzó valores extremos de hasta 31.1 m/s, aunque la mayoría de los registros fueron considerablemente más bajos.

Las variables categóricas también fueron exploradas. La condición de cielo ('cielo_visible') indicó que en un 62.7% de los registros se reportó cielo nublado, lo cual es consistente con las características de una isla expuesta a masas de aire oceánico. La distribución por mes y estación fue relativamente homogénea, aunque se identificó una ligera concentración de registros durante el otoño (27.2%), lo que podría tener implicancias en la estacionalidad de las precipitaciones. Para facilitar análisis posteriores, se creó la variable binaria 'lluvia', la cual permite discriminar eventos

de precipitación con base en un umbral relevante, así como la variable 'periodo', que agrupa los registros por franjas horarias según el momento del día.

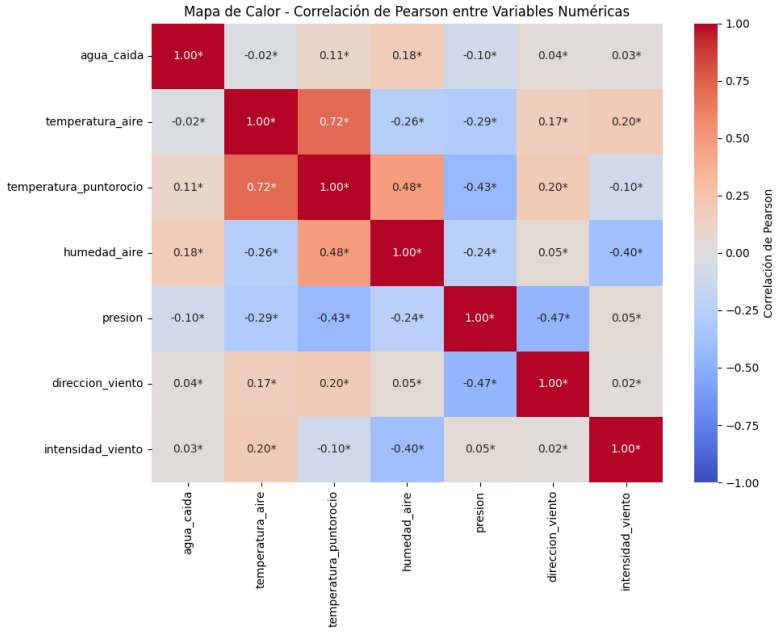
En el análisis bivariado, se analizaron diferencias entre observaciones con y sin lluvia. Por ejemplo, la mediana de 'agua_caida' fue 1.2 mm/h en registros con lluvia, frente a 0 mm/h cuando no llovió, lo que valida la segmentación binaria. Asimismo, se observó que la 'temperatura_aire' fue ligeramente menor durante lluvia (mediana = 20.15 °C), mientras que la 'temperatura_puntorocio' fue superior (mediana = 18.71 °C), lo cual sugiere un mayor grado de saturación del aire en presencia de precipitaciones. La variable 'humedad_aire' presentó un incremento notable durante lluvia (mediana = 91.46%) comparado con condiciones secas (mediana = 76.2%). En tanto, la presión atmosférica fue, en promedio, algo menor durante lluvia, lo que puede vincularse con inestabilidad atmosférica. Las variables de viento, por otro lado, no mostraron diferencias sustanciales entre ambos escenarios, reafirmando su limitada utilidad predictiva.



En el análisis multivariado exploratorio se calcularon las correlaciones lineales entre las variables cuantitativas. Las correlaciones más relevantes con la variable dependiente 'agua_caida' fueron:

- 'humedad_aire': Correlación positiva leve (0.18)
- 'temperatura_puntorocio': Correlación positiva leve (0.11)
- 'presion': Correlación negativa leve (-0.10)

Aunque las correlaciones individuales con la variable objetivo son bajas, su combinación puede aportar poder predictivo al integrarlas en modelos multivariados. Además, se identificaron relaciones significativas entre las variables explicativas: la correlación entre 'temperatura_aire' y 'temperatura_puntorocio' fue alta (0.72), lo que refleja su conexión física directa; mientras que se observaron correlaciones negativas entre 'temperatura_aire' y 'humedad_aire' (-0.26), así como entre 'temperatura_puntorocio' y 'presion' (-0.43). Estas asociaciones reflejan interacciones climatológicas relevantes entre temperatura, humedad y presión, y permiten inferir dinámicas atmosféricas útiles para el modelado de precipitaciones.



A partir de este segundo análisis exploratorio se pueden destacar los siguientes hallazgos preliminares:

- La lluvia en Rapa Nui es un fenómeno poco frecuente, pero cuando ocurre, puede presentarse con alta intensidad. Este desbalance entre clases debe ser considerado en el diseño del modelo de clasificación mediante el uso de métricas robustas ante clases desbalanceadas, como la 'balanced_accuracy'.
- Las variables 'temperatura_puntorocio', 'humedad_aire', 'presion' y 'cielo_visible' demostraron ser las más relevantes para discriminar entre condiciones de lluvia y no lluvia, por lo que se consideraron prioritarias en el diseño del modelo de clasificación.
- Se evidencia cierta estacionalidad en los eventos lluviosos, con una mayor ocurrencia al finalizar el verano e iniciar el otoño. Esto refuerza la inclusión de la variable 'mes' como componente explicativo.
- La eliminación de las variables relacionadas con el viento resultó adecuada, ya que presentaban baja correlación con la variable objetivo y un comportamiento altamente disperso, lo que reducía su valor predictivo.
- La aplicación del método de imputación por vecinos más cercanos (KNN) permitió completar los registros faltantes de forma coherente, utilizando información del espacio multidimensional definido por las variables disponibles. En paralelo, la normalización de los datos garantizó la compatibilidad entre escalas, facilitando el entrenamiento de modelos basados en distancia o árboles de decisión.

Chapter 4

Metodología

Los datos utilizados en este análisis provienen de la estación meteorológica Mataveri, ubicada en Isla de Pascua. La base original contenía registros por minuto de las siguientes variables: Agua caída (mm/h), presión al nivel del mar (hPa), humedad relativa del aire (%), temperatura del aire seco (°C), temperatura del punto de rocío (°C), intensidad promedio del viento en el instante de medición (kt), dirección promedio del viento (°), y condición de cielo visible (1: despejado, 0: nublado), además de la fecha y momento de medición.

Debido a la alta frecuencia de los datos (por minutos), se procedió a una transformación para agrupar los registros a una frecuencia horaria. Esta transformación se implementó en RStudio, utilizando los siguientes criterios:

- Agua caída: Se calculó como la suma acumulada por hora
- Presión, humedad, temperatura del aire, temperatura del punto de rocío, dirección e intensidad del viento: Se calcularon como sus respectivos promedios por hora.
- Condición de cielo visible: Se definió como 'despejado' (valor 1) si al menos un 10% de los minutos dentro de la hora tenían esa condición, de lo contrario se consideró 'nublado' (valor 0)

Durante la exploración de los datos, se identificó la presencia de valores faltantes, especialmente en variables como agua caída, presión, humedad relativa y condición de cielo visible. Para abordar este problema, se utilizó la técnica de imputación por vecinos más cercanos (K-Nearest Neighbors, KNN), implementada en Python. Este método estima los valores faltantes en función de la similitud con observaciones completas, calculando distancias euclidianas en un espacio definido por variables relevantes y correlacionadas con la variable a imputar (**?**).

Para seleccionar el número óptimo de vecinos (k), se empleó un enfoque de validación cruzada utilizando el 20% de los registros completos y evaluando el error cuadrático medio (RMSE) para diferentes valores de k. Se eligió aquel que minimizó el RMSE en cada caso. Para la imputación de variables numéricas se utilizó el promedio de los valores de los vecinos seleccionados, mientras que para las variables categóricas se aplicó la moda.

Este procedimiento no requiere suposiciones distribucionales ni el ajuste de modelos paramétricos, ya que se basa únicamente en la similitud local entre observaciones. Según un estudio reciente, KNN ha demostrado un buen desempeño en la imputación de datos en series temporales multivariadas, posicionándose como uno de los métodos más eficaces después de enfoques complejos basados en aprendizaje profundo (**? ?**).

Por último, dado que las variables del conjunto de datos presentan diferentes escalas, se normalizaron las variables numéricas utilizando el método MinMaxScaler, que transforma los valores al rango [0,1]. Esta transformación favorece el rendimiento del modelo XGBoost al asegurar una escala uniforme, especialmente útil cuando se utilizan algoritmos sensibles a las magnitudes. Además, las variables categóricas fueron transformadas mediante codificación one-hot (dummies), con el fin de integrarlas adecuadamente en el modelo de clasificación.

Para la construcción y selección del modelo predictivo se empleó XGBoost (Extreme Gradient Boosting), una técnica de aprendizaje supervisado basada en 'gradient boosting', propuesta inicialmente por Friedman (2001) (8) y optimizada por Chen y Guestrin (2016) (1). Este algoritmo genera árboles de decisión de forma secuencial, donde cada nuevo árbol se ajusta para corregir los errores residuales del conjunto de árboles anteriores, utilizando un enfoque de optimización de gradientes. Este método incorpora términos de regularización L1 y L2, los cuales controlan la complejidad del modelo y reducen el riesgo de sobre ajuste, lo cual es particularmente relevante al modelar fenómenos meteorológicos como la lluvia, que presentan alta variabilidad y no linealidad (1).

Gracias a su enfoque iterativo y capacidad para manejar relaciones no lineales e interacciones entre múltiples variables, XGBoost es especialmente útil para capturar patrones complejos presentes en los datos climáticos. Variables como temperatura del aire, punto de rocío, humedad relativa y presión atmosférica pueden influir de manera conjunta y no lineal en la ocurrencia de lluvia. Estudios recientes destacan su buen desempeño en este tipo de aplicaciones, mostrando una elevada capacidad predictiva frente a otros algoritmos (2; 12).

XGBoost utiliza conjuntos de árboles de decisión como modelos base. Cada modelo individual f_k corresponde a un árbol de regresión, y la predicción final para un registro se obtiene como la suma de las predicciones individuales de todos los árboles entrenados. Este enfoque de ensamblado permite construir un predictor robusto a partir de modelos débiles.

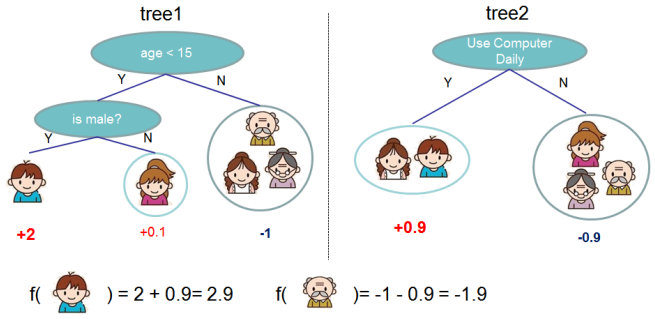


Figure 4.1: Esquema de XGBoost (adaptado de Chen & Guestrin, 2016).

En la figura anterior se observa cómo cada árbol aporta un valor predictivo individual, y la salida total es la suma acumulada de estos valores. A diferencia de un solo árbol de decisión, donde se toma una única ruta para hacer la predicción, XGBoost construye múltiples árboles en secuencia, donde cada árbol trata de minimizar los errores cometidos por los árboles anteriores. Este procedimiento continúa hasta alcanzar un número máximo de iteraciones o hasta que las mejoras sean insignificantes.

En términos generales, el modelo XGBoost se fundamenta en dos ideas principales: el ensamblaje de árboles y el gradient boosting. El primero refiere al uso de múltiples árboles de decisión o regresión como modelos base, mientras que el segundo corresponde a la construcción secuencial de estos árboles, donde cada uno corrige los errores cometidos por los anteriores a través de la minimización de una función de pérdida.

Cada árbol $f_t(x_i)$ en el modelo se entrena para corregir los errores residuales de la predicción acumulada hasta el árbol $t-1$. Matemáticamente, si $\hat{y}_i^{(t-1)}$ es la predicción para la observación i hasta la iteración $t-1$, el nuevo árbol $f_t(x_i)$ ajusta esta predicción según:

$$\hat{y_i}^{(t)} = \hat{y_i}^{(t-1)} + f_t(x_i)$$

Al finalizar las iteraciones, la predicción total es la suma de los aportes de todos los árboles:

$$\hat{y_i} = \sum_{k=1}^{T} \left(f_k(x_i)\right)$$

Este procedimiento fue detallado por Chen y Guestrin (2016) en su publicación "XGBoost: A Scalable Tree Boosting System" (KDD 2016), y ha sido ampliamente adoptado en múltiples lenguajes de programación por su eficiencia y precisión (1).

El entrenamiento de este modelo sigue en esencia el esquema de Gradient Boosting, donde como pasos clave está en el uso de la expansión de Taylor de segundo orden para la optimización, y también agregando un componente de regularización de los árboles. Generalizando, el algoritmo recorre iterativamente etapas:

1. Inicialización: El modelo parte con una predicción constante inicial, usualmente igual al promedio de la variable dependiente. Esta predicción base se conoce como base_score, y corresponde al árbol 0 del modelo.

2. Cálculo de gradientes y Hessianos: Para cada muestra i, se calculan el gradiente g_i y el Hessiano h_i (segunda derivada) de la función de pérdida l con respecto a la predicción actual $\hat{y}_i^{(t-1)}$:

$$g_i = \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i}, \quad h_i = \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^2}$$

Estos valores muestran cómo en cada muestra cambiaría la pérdida al ajustar la predicción, y son usadas para entrenar el nuevo árbol.

3. Construcción de un nuevo árbol f_t: Con los valores (x_i, g_i, h_i), se construye un árbol de regresión que minimiza de forma aproximada la función objetivo. El árbol se expande de forma codiciosa, seleccionando en cada nodo la división que maximiza la reducción de pérdida. La calidad de una división se evalúa mediante la ganancia (Gain) dada por la fórmula (Eq. 7 en Chen & Guestrin):

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma$$

donde G_L, H_L, G_R y H_R son las sumas de gradientes y Hessianos en las ramas izquierda y derecha, respectivamente. Los términos λ y γ son hiperparámetros de regularización: λ penaliza pesos grandes (regularización L2) y γ penaliza el número de hojas.

La división solo se realiza si la ganancia es positiva. El crecimiento del árbol se detiene si se alcanzan límites como profundidad máxima, ganancia mínima o número mínimo de muestras por hoja (13).

4. Cálculo de pesos de las hojas: Una vez definida la estructura del árbol, se calcula el peso óptimo w_j para cada hoja j minimizando la función objetivo aproximada. Debido a la formulación cuadrática gracias al uso de los g_i, h_i, donde el peso óptimo de la hoja j es:

$$w_j = -\frac{G_j}{H_j+\lambda}$$

donde $G_j = \sum_{i\in I_j} g_i$ y $H_j = \sum_{i\in I_j} h_i$ son las sumas de gradientes y Hessianos de las instancias en la hoja j. Esta fórmula surge del mínimo de la ecuación cuadrática y depende de la regularización L2 λ. También, se puede calcular la ganancia total asociada al árbol completo:

$$Gain_{tree}(obj) = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda} + \gamma T$$

donde T es el número de hojas (cada hoja suma un término γ al coste). Esta medida permite comparar diferentes estructuras de árbol.

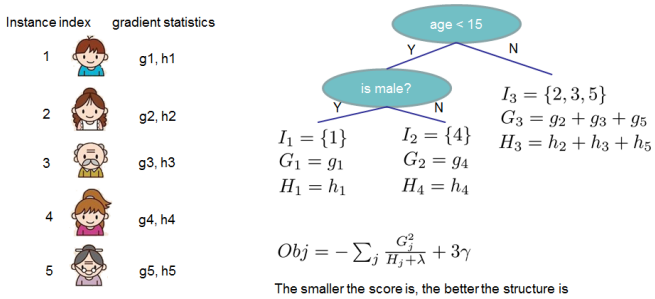


Figure 4.2: Ejemplo de árbol de decisión (adaptado de Chen & Guestrin, 2016).

5. Aplicación del árbol: El árbol f_t entrenado se escala mediante la tasa de aprendizaje η (eta) para controlar su influencia. Las predicciones de este árbol se multiplican por η (por defecto 0.3). Este paso es conocido como 'shrinkage', reduciendo el sobre ajuste al disminuir la contribución de cada árbol individual. Después, se actualizan las predicciones acumuladas:

$$\hat{y}_i^t = \hat{y}_i^{(t-1)} + \eta f_t(x_i)$$

6. Iteración y parada: Se repite este proceso (cálculo de gradientes, construcción del árbol y actualización) por un número predefinido de rondas (boosting rounds) o hasta un criterio de parada. Cada iteración incorpora un nuevo árbol que ajusta los errores de los anteriores.

Finalmente, este proceso iterativo de optimización y minimización de la función objetivo garantiza un adecuado ajuste a los datos, permitiendo al modelo aprender de los errores cometidos en etapas previas. A su vez, la inclusión de términos de regularización (como λ y γ) previene el sobre ajuste, limitando la complejidad excesiva de los árboles de decisión y estabilizando los pesos asignados en cada hoja.

El modelo XGBoost es un marco general basado en Gradient Boosting de árboles de decisión, aplicable tanto a tareas de regresión como de clasificación. La principal diferencia entre estos enfoques radica en la elección de la función de pérdida y en la naturaleza de la salida generada. Para problemas de regresión, se emplea típicamente la función de pérdida del error cuadrático medio (reg:squarederror), generando predicciones numéricas continuas. En cambio, para clasificación binaria, se utiliza una pérdida logística (log-loss), lo que permite obtener una probabilidad de pertenencia a una clase y se aplica una función sigmoide en la salida final (binary:logistic).

Diversas investigaciones han demostrado que XGBoost supera en rendimiento a otros algoritmos de aprendizaje automático como Random Forest o incluso modelos más complejos como LSTM, especialmente en la predicción de eventos extremos relacionados con precipitaciones. Por ejemplo, el estudio de Kumar et al. (2025) reportó un error cuadrático medio de apenas 0,12 al predecir eventos tipo cloudburst, lo cual validó su eficacia dentro de sistemas de alerta temprana. Este tipo de evidencia respalda la selección de XGBoost como modelo apropiado para nuestro objetivo de predecir la ocurrencia de lluvia en intervalos horarios, dado que ofrece un equilibrio entre robustez predictiva y capacidad de modelar relaciones climáticas no lineales.

Para implementar el modelo de clasificación binaria, se construyó una variable objetivo llamada 'Lluvia', donde se codifica 1 si se considera que llueve y 0 si no. Esta variable se define a partir de un umbral aplicado a la variable 'agua_caida' (agua caída acumulada por hora, en mm). Si bien la Organización Meteorológica Mundial (WMO) no establece un umbral universal fijo para diferenciar llovizna de lluvia, sí clasifica la llovizna según el tamaño de las gotas ($<$0.5 mm) y sugiere evaluaciones basadas en la intensidad horaria.

En esta línea, el Manual conjunto de la OACI y la WMO (2021) considera como llovizna cualquier precipitación menor a 0.1 mm/h, y a partir de esa tasa, la cataloga como lluvia ligera. Por su parte, Gultepe (2008) señala que valores menores o iguales a 0.3 mm/h suelen asociarse a lloviznas densas o nieblas, lo que coincide con el criterio operativo adoptado por la American Meteorological Society (AMS, s.f.), que ubica el umbral de llovizna en aproximadamente 0.3 mm/h.

Dado el contexto geográfico del estudio, la Isla de Pascua, ubicada en una zona costera con alta humedad relativa y fenómenos microclimáticos propios de regiones marítimas, se decidió establecer un umbral más conservador. En consecuencia, se considerará que existe lluvia (valor 1) cuando el registro de agua caída por hora supere los 0.4 mm/h, lo cual representa un criterio más estricto que la media de la literatura, pero adecuado para evitar falsas clasificaciones en un entorno donde las lloviznas pueden ser frecuentes pero poco significativas en términos de acumulación hídrica.

Este criterio permitirá al modelo distinguir de manera más precisa entre episodios con y sin lluvia real, mejorando la calidad de la clasificación binaria y optimizando la función de pérdida logística empleada en el entrenamiento de XGBoost.

Las principales librerías utilizadas en este estudio, junto con sus respectivas funciones, fueron:

- **NumPy**: Utilizada para realizar operaciones numéricas y manejo de arreglos multidimensionales, facilitando cálculos y transformaciones de datos.
- **Pandas**: Empleada para la carga, limpieza, transformación y agrupación de los datos meteorológicos en estructuras tipo data frame.
- **Scikit-learn**:
  - 'KNNImputer': Para la imputación de valores faltantes mediante vecinos más cercanos.
  - 'MinMaxScaler': Para escalar variables numéricas al rango [0, 1].
  - 'train_test_split': División del conjunto de datos en entrenamiento y prueba.
  - 'RandomizedSearchCV': Optimización de hiperparámetros mediante búsqueda aleatoria.
  - 'StratifiedKFold' y 'KFold': Para realizar validación cruzada estratificada y simple.
  - 'mean_squared_error' y 'r2_score': Métricas de evaluación para modelos de regresión.
  - 'confusion_matrix' y 'ConfusionMatrixDisplay': Evaluación del desempeño del modelo de clasificación.
- **XGBoost**: Librería principal utilizada para construir, entrenar y evaluar el modelo de clasificación basado en gradient boosting.

- **Seaborn** y **Matplotlib**: Utilizadas para visualización de datos, gráficos exploratorios, análisis de correlaciones y resultados del modelo.
- **JSON**: Para la manipulación de archivos estructurados, almacenamiento de configuraciones y resultados.

Chapter 5

Desarrollo y evaluación del modelo

Luego de determinar las variables óptimas para la predicción tanto de lluvia (1: lluvia, 0: no lluvia) como de agua caída (mm/h): Siendo las variables óptimas Temperatura promedio del punto de rocío (°C), Humedad promedio relativa (%), Presión promedio al nivel del mar (hPa), Mes (Enero, ..., Diciembre), Periodo (Madrugada, Mañana, Tarde, Noche) y Condición de cielo visible (1: Despejado, 0: Nublado).

Al verificar la existencia de valores faltantes, antes de comenzar con el modelo se determinó la imputación por vecinos más cercanos (KNN). Donde, las variables y cuantos valores a estimar sus registros faltantes son: Agua caída en una hora (12 observaciones), Presión promedio a nivel del mar (9 observaciones), Condición de cielo visible (12 observaciones) (parámetros definidos en 3.1). Guardando estos valores estimados por los valores faltantes, quedando una base de datos completa y final.

Con las variables ya transformadas con MinMaxScaler para variables numéricas (Temperatura promedio del punto de rocío, Humedad promedio relativa, Presión promedio al nivel del mar) y aplicando la separación de las variables categóricas como dummies (Mes, Periodo y Condición de cielo visible) para la correcta interpretación de estas variables para establecer el modelo. Con esto, es posible seguir de manera válida la definición del modelo.

5.1 Modelo de clasificación XGBoost

Con la variable creada "lluvia" establecida como (1: lluvia, 0: no lluvia), donde si agua caída acumulada en una hora era mayor a 0.4 mm/h se clasifica como lluvia, si es menor o igual a 0.4 mm/h se clasifica como no lluvia (llovizna). Se definirá un modelo para clasificar la lluvia.

Como primer paso, se realizó la división entre entrenamiento (80% de las observaciones) y validación (20% de las observaciones), sin mezcla y respetando el orden temporal. Para reproducibilidad se estableció un random_state=42 (semilla). Luego para calcular que tan desbalanceadas están las clases se define un peso que el modelo XGBoost utiliza para penalizar los errores sobre la clase minoritaria, se realiza un conteo de las clases (1: lluvia, 0: no lluvia) con el fin de calcular el peso (scale_pos_weight).

$$scale\_pos\_weight = \frac{\text{número de horas sin lluvia en entrenamiento}}{max(1, \text{número de horas con lluvia en entrenamiento})}$$

Se establece XGBoost como un modelo de clasificación con función objetivo (binary:logistic) y métrica de evaluación AUC (área bajo la curva ROC) con resultados óptimos en clases desbalanceadas y se establece un random_state = 42 para reproducibilidad.

Table 5.1: Parámetros del modelo `XGBClassifier` (base_clf)

Parámetro	Valor
objective	'binary:logistic'
eval_metric	'auc'
random_state	42

El espacio de búsqueda de los hiperparámetros óptimos son:

Table 5.2: Distribución de parámetros para la búsqueda aleatoria del modelo XGBClassifier (param_dist)

Parámetro	Valores
n_estimators	{100, 200, 300}
max_depth	{3, 5, 7}
learning_rate	{0.01, 0.05, 0.1}
subsample	{0.6, 0.8, 1.0}
colsample_bytree	{0.6, 0.8, 1.0}
gamma	{0, 1, 5}
scale_pos_weight	{1, scale_pos_weight}

También se configura la validación cruzada usando 5 folds (divisiones de entrenamiento). Este proceso por cada iteración entrena el modelo 5 veces, donde: En la primera vez lo entrena con las particiones 1-4 y válida con el 5 fold, luego lo entrena con 1,2,3,5 fold y válida con la cuarta partición. Hasta cubrir todas las combinaciones. Finalmente, calculando el promedio del rendimiento con balanced_accuracy (random_state=42) Esto es relevante debido a que asegura que el modelo es válido para la totalidad de los datos y no solo los últimos 20% de la validación.

Se define una función de búsqueda, donde se introducen lo expuesto anteriormente con 30 combinaciones de hiperparámetros (elegidos aleatoriamente), pero con un random_state=42 para reproducibilidad.

Table 5.3: Parámetros usados en `RandomizedSearchCV` para modelo XGBClassifier

Parámetro	Valor
estimator	base_clf
param_distributions	param_dist
n_iter	30
scoring	'balanced_accuracy'
cv	cv
n_jobs	-1
random_state	42
verbose	1
refit	True

Finalmente, se realiza la búsqueda encontrando la siguiente combinación de hiperparámetros con un balanced_accuracy de 0.854 (indicando que el modelo detecta el 85.4% de las veces cuando predice lluvia o no lluvia). Estos hiperparámetros serán utilizados para entrenar el modelo XGBoost para clasificación final.

Table 5.4: Parámetros óptimos seleccionados por `RandomizedSearchCV` para modelo XGBClassifier (best_params)

Parámetro	Valor óptimo
n_estimators	200
max_depth	3
learning_rate	0.05
subsample	0.6
colsample_bytree	1.0
gamma	0
scale_pos_weight	24.051

Usando los mejores parámetros encontrados y definidos anteriormente, definimos el modelo de clasificación XGBoost final con la métrica de evaluación AUC.

Table 5.5: Parámetros del modelo `final_clf` con `XGBClassifier`

Parámetro	Valor
params	(valores óptimos encontrados best_params)
objective	'binary:logistic'
eval_metric	'auc'
use_label_encoder	False
random_state	42

Se entrena el modelo final con todos los datos de entrenamiento (80%) y realiza la clasificación de lluvia con base en las variables predictoras de validación (Temperatura del punto de rocío, Humedad relativa, Presión, Mes, Periodo y Condición del cielo visible) para la validación de la variable lluvia (20%). Y con ello, calcular la matriz de confusión. Entregando los siguientes resultados.

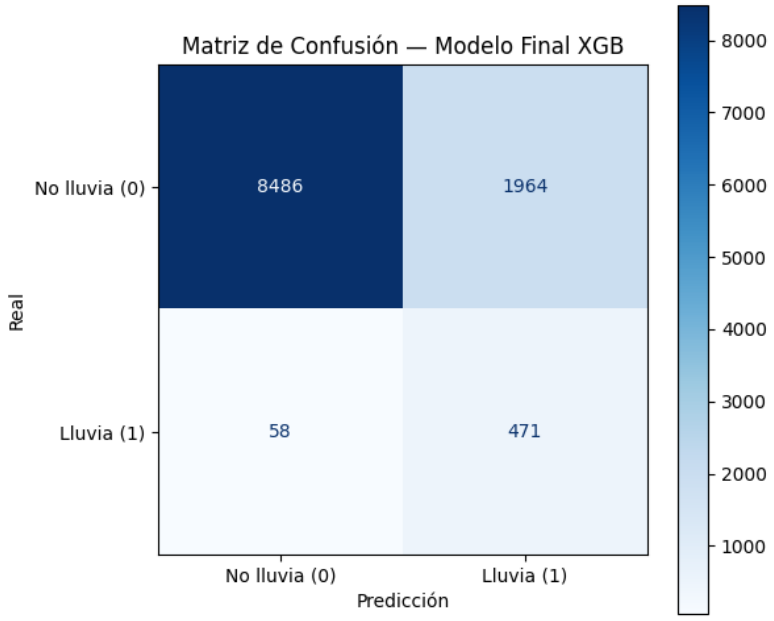


Figure 5.1: Matriz de confusión: Modelo de clasificación XGBoost para lluvia

Interpretando esta matriz de confusión, se observa que el modelo clasifica correctamente la clase 'no lluvia' en 8.486 ocasiones (81.2% de las horas sin lluvia), mientras que se equivoca en 58 ocasiones (0.68% de las horas con y sin lluvia). Respecto a la clase "lluvia", el modelo acierta en 471 casos (89% de las horas con lluvia), pero falla en 1.964 casos (81% de las horas con y sin lluvia).

Viendo esto, podemos determinar el equilibrado rendimiento del modelo de clasificación, donde en ambos casos (clasificación correcta de lluvia y no lluvia) superan el 80% de los casos en sus clases. A pesar de conseguir el modelo óptimo para clases desbalanceadas, sigue con error al predecir lluvia, donde cuando predice lluvia (entre cuando realmente llueve y no llueve) solo acierta el 19% de las veces, pero cuando predice que no llueve (entre cuando realmente llueve y no llueve) acierta el 99% de las veces.

Este modelo capturando correctamente los eventos en los que llueve entre los casos en los que llueve y manteniendo un balance en las clasificaciones (a pesar de ser una variable altamente desbalanceada).

Este modelo es mejorable para su aplicación como alerta temprana. Sin embargo, requiere el respaldo de variables pronosticadas de otros modelos (humedad, presión, cielo visible y temperatura a punto de rocío) donde es posible obtener de modelos precisos y robustos ya definidos, los cuales utilizan variables avanzadas para contrarrestar la complejidad de las variables meteorológicas (variables satelitales, anomalías, etc.)

5.2 Modelo de regresión XGBoost

En este caso, validaremos un modelo de regresión definido para agua caída con XGBoost. La métrica utilizada para ir seleccionando el mejor modelo de cierta iteración y con la que seguirá optimizando será la raíz del error cuadrático medio. Buscando los mejores parámetros para luego ajustar el modelo (con las mismas variables anteriormente definidas como predictoras) y luego mostrar una serie de tiempo en las horas de validación del modelo.

Como primer paso, se realizó la división entre entrenamiento (80% de las observaciones) y validación (20% de las observaciones), sin mezcla y respetando el orden temporal. Para reproducibilidad se estableció un random_state=42 (semilla).

Se establece XGBoost como un modelo de regresión para agua caída con función objetivo (reg:squarederror) y se define un random_state = 42 para reproducibilidad.

Table 5.6: Parámetros del modelo `XGBRegressor` (base_reg)

Parámetro	Valor
objective	'reg:squarederror'
random_state	42

El espacio de búsqueda de los hiperparámetros óptimos son:

Table 5.7: Distribución de parámetros para la búsqueda aleatoria del modelo `XGBRegressor` (`param_dist`)

Parámetro	Valores
n_estimators	{100, 200, 300, 500}
max_depth	{3, 5, 7, 9}
learning_rate	{0.001, 0.01, 0.05, 0.1}
subsample	{0.6, 0.8, 1.0}
colsample_bytree	{0.6, 0.8, 1.0}
gamma	{0, 1, 5, 10}
reg_alpha	{0, 0.1, 1}
reg_lambda	{1, 5, 10}

También se configura la validación cruzada usando 5 folds (divisiones de entrenamiento). El procedimiento de este es el mismo definido anteriormente. Finalmente, calculando el promedio del rendimiento entre estas validaciones con negativa raíz cuadrática del error (RMSE negativo), negativo debido a que lo definido en el modelo de búsqueda maximiza, por lo que al dejarlo negativo nos minimizara el RMSE (para reproducibilidad random_state=42). Estas validaciones son relevantes debido a que asegura que el modelo es válido para la totalidad de los datos y no solo los últimos 20% de la validación.

Se define la función de búsqueda, donde se introducen lo expuesto anteriormente con 30 combinaciones de hiperparámetros (elegidos aleatoriamente), pero con un random_state=42 para reproducibilidad.

Table 5.8: Parámetros usados en `RandomizedSearchCV` para el modelo `XGBRegressor`

Parámetro	Valor
estimator	base_reg
param_distributions	param_dist
n_iter	30
scoring	'neg_root_mean_squared_error'
cv	cv
n_jobs	-1
random_state	42
verbose	1
refit	True

Finalmente, se realiza la búsqueda encontrando la siguiente combinación de hiperparámetros con un RMSE de 0.5986 (indicando que el modelo al predecir incurre en una raíz del error cuadrático medio de 0.5986 mm/h). Estos hiperparámetros serán utilizados para entrenar el modelo XGBoost para regresión final.

Table 5.9: Parámetros óptimos seleccionados por `RandomizedSearchCV` para el modelo `XGBRegressor` (`best_params`)

Parámetro	Valor óptimo
`n_estimators`	200
`max_depth`	5
`learning_rate`	0.05
`subsample`	1.0
`colsample_bytree`	0.6
`gamma`	1
`reg_alpha`	1
`reg_lambda`	1

Usando los mejores parámetros encontrados y definidos anteriormente, definimos el modelo de regresión XGBoost final con la métrica de evaluación raíz cuadrática del error (squarederror).

Table 5.10: Parámetros del modelo `final_clf` con `XGBRegressor`

Parámetro	Valor
`params`	(valores óptimos encontrados `best_params`)
`objective`	`'reg:squarederror'`
`eval_metric`	`'rmse'`
`use_label_encoder`	`False`
`random_state`	42

Finalmente, se entrena el modelo final con todos los datos de entrenamiento (80%) y realiza la predicción de agua caída con base en las variables predictoras de validación (Temperatura del punto de rocío, Humedad relativa, Presión, Mes, Periodo y Condición del cielo visible) para la validación de agua caída (20%). Y con ello, revisar las predicciones del agua caída real con la pronosticada en una serie temporal. Entregando lo siguiente:

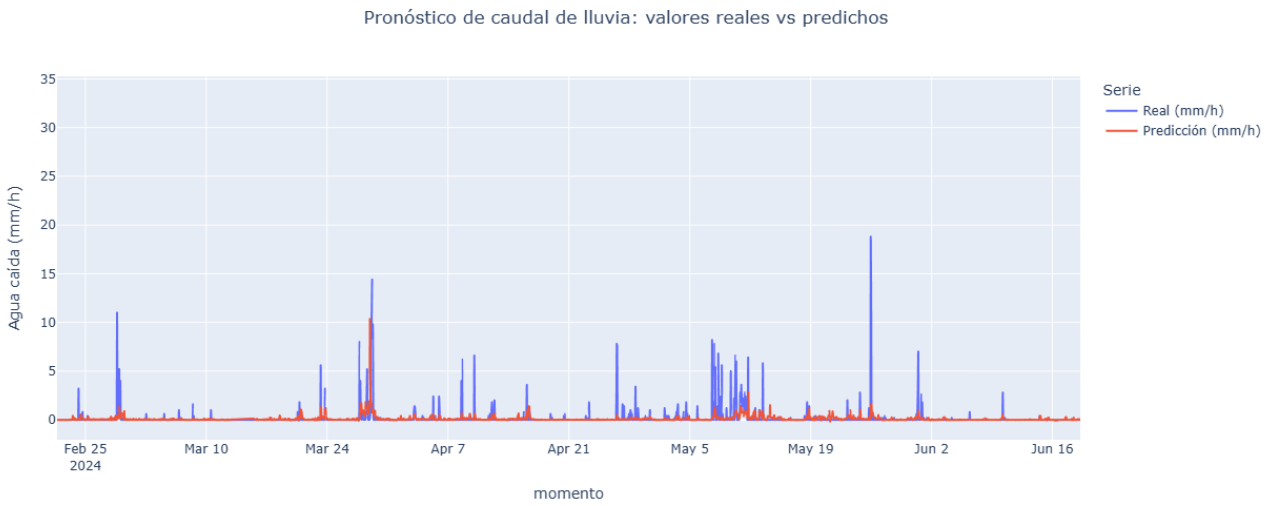


Figure 5.2: Serie temporal parte 1: Validación de la predicción del modelo XGBRegressor para el agua caída entre febrero y junio de 2024

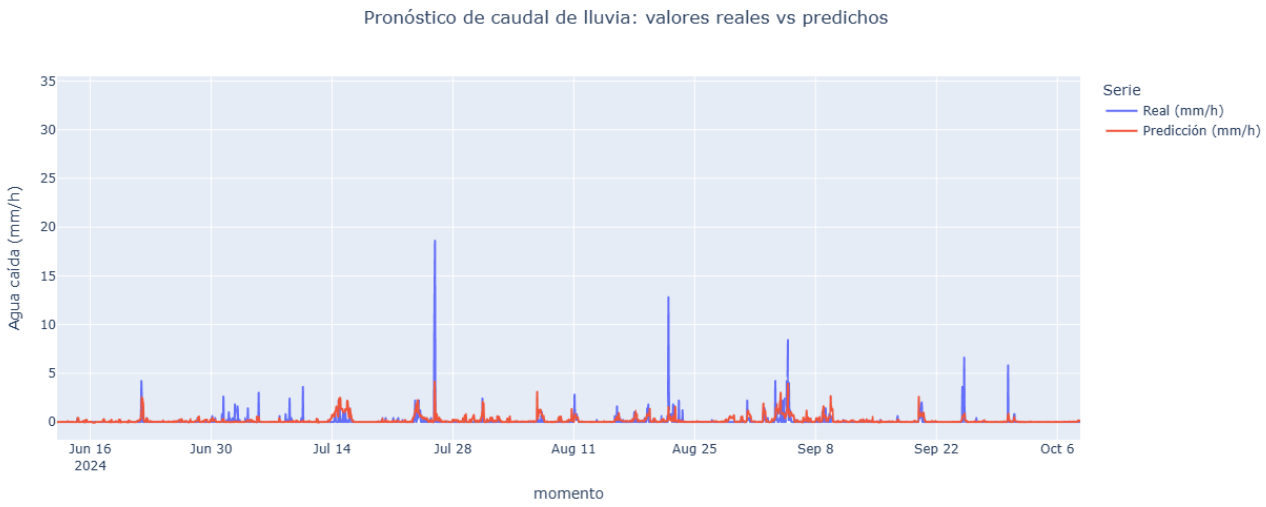


Figure 5.3: Serie temporal parte 2:Validación de la predicción del modelo XGBRegressor para el agua caída entre junio y octubre de 2024

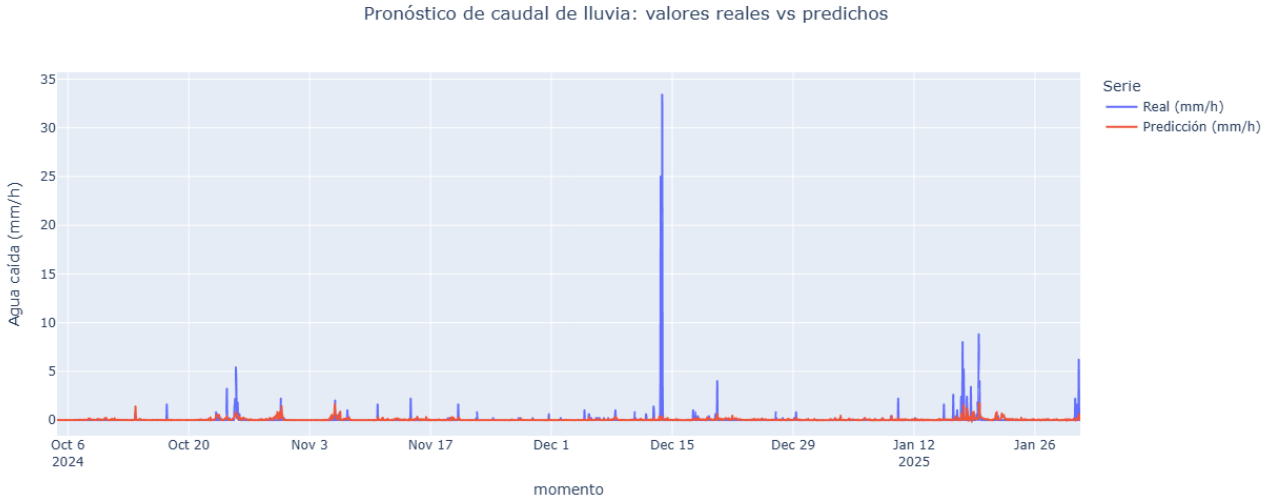


Figure 5.4: Serie temporal parte 3: Validación de la predicción del modelo XGBRegressor para el agua caída entre octubre de 2024 y enero de 2025

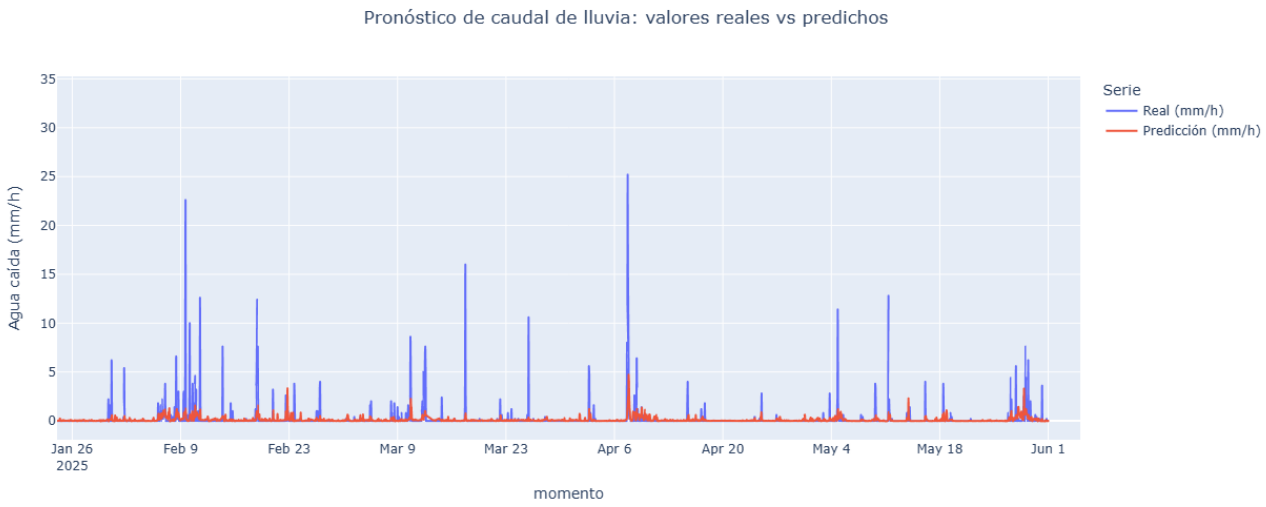


Figure 5.5: Serie temporal parte 4: Validación de la predicción del modelo XGBRegressor para el agua caída entre enero y junio de 2025

En general, viendo esta serie temporal de valores reales y pronosticados de agua caída podemos notar que los valores reales en las horas de validación mantienen diferentes momentos donde el agua caída o precipitaciones son de un momento a otro de gran magnitud, llegando en periodos cortos de tiempo (una hora) a 25 mm/h en abril de 2025 o bien más 30 mm/h en diciembre de 2024. Donde, la predicción en la mayoría detecta los momentos de lluvia. No obstante, no logra capturar la alta variabilidad y magnitud presente en las precipitaciones reales (subestimando). El modelo definido entregó las siguientes métricas en la validación.

Table 5.11: Desempeño en conjunto de prueba para XGBRegressor

Métrica	**Valor**
RMSE Test	0.8903
R^2 Test	0.1161

Con estas métricas también podemos notar el pobre rendimiento del modelo para explicar la variabilidad del agua caída ($R^2 = 0.1161$) e incurriendo en una alta raíz cuadrática del error ($RMSE = 0.8903$).

Este comportamiento visto sugiere que, aunque al base de variables actuales aporta información relevante sobre la ocurrencia en mm/h de precipitaciones, no captura por completo los valores extremos que generan momentos de lluvia intensa. Para mejorar la precisión de los volúmenes estimados, sería necesario recolectar (si es posible) algunas variables extra como indicadores de anomalías climáticas o datos satelitales de densidad de nubes.

Por lo tanto, al determinar que es necesario agregar otras variables para cuantificar la cantidad de lluvia que caerá correctamente, concluimos en que el modelo de regresión para agua caída detecta correctamente los momentos de lluvia, pero con un gran margen de mejora al agregar tales variables. Generalizando esta conclusión para ambos modelos definidos.

Chapter 6

Resultados, conclusiones y recomendaciones

6.1 Resultados

Las variables empleadas para los modelos tanto para lluvia como agua caída fueron las siguientes:

- Continuas: Humedad relativa del aire, temperatura a punto de rocío y presión a nivel del mar
- Categóricas: Condición de cielo visible, mes del año, periodo del día (mañana, tarde, noche, madrugada)

Las 12 (agua_caida), 9 (presion) y 12 (cielo_visible) valores faltantes estimados por KNN (vecinos más cercanos) se encuentran expuestos en la tabla 3.2, y su estimación fue por 15, 10 y 15 k-vecinos respectivamente.

Para el modelo de clasificación XGBoost, la búsqueda aleatoria de hiperparámetros alcanzó un balanced_accuracy de 0.854 en validación cruzada, seleccionando los siguientes parámetros óptimos: $n\_estimators = 200$, $max\_depth = 3$, $learning\_rate = 0.05$, $subsample = 0.6$, $colsample_bytree = 1.0$, $gamma = 0$, $scale_pos_weight = 24.05$.

Al validar el modelo de clasificación se obtuvieron los siguientes resultados:

- Lluvia correctamente clasificada (TP): 471 (89% de las horas con lluvia)
- Lluvia real no detectada (FN): 58 (11% de las horas con lluvia)
- Sin lluvia correctamente clasificada (TN): 8486 (81.2% de los casos sin lluvia)
- Sin lluvia real no detectada (FP): 1964 (18.8% de los casos sin lluvia)

A pesar de ser el modelo de clasificación óptimo según los criterios evaluados, presenta un desempeño deficiente al predecir eventos de lluvia, ya que solo acierta en el 19% de los casos en que pronostica lluvia, lo que evidencia una alta tasa de falsos positivos.

En el modelo de regresión XGBoost la optimización sobre la métrica RMSE (neg_root_mean_squared_error) seleccionó los siguientes parámetros: $n\_estimators = 200$, $max\_depth = 5$, $learning\_rate = 0.05$, $subsample = 1.0$, $colsample_bytree = 0.6$, $gamma = 1$, $reg_alpha = 1$, $reg_lambda = 1$.

En la validación cruzada el RMSE promedio fue de 0.5986 mm/h, pero al evaluar en el conjunto de validación se obtuvo un RMSE 0.8903 mm/h y un R^2 de solo 0.1161, lo que evidencia la baja explicación de la variabilidad de los pronósticos de lluvia y una tendencia a subestimar los puntos altos de precipitación.

La serie temporal de predicciones vs valores reales de agua caída de igual manera muestra que, aunque el modelo detecta correctamente momentos de lluvia, la magnitud predicha casi nunca alcanza los valores observados, especialmente en eventos de precipitaciones intensa.

6.2 Conclusiones

Los patrones climáticos observados al analizar la distribución de lluvia muestra una media de 0.106 mm/h y un máximo de precipitaciones en 33.4 mm/h. En cuanto a su estacionalidad, se hallaron mayores frecuencias de lluvia a finales de verano e inicios de otoño.

Se identificaron variables clave a través de análisis bivariado y correlaciones, las más relevantes para agua caída fueron humedad relativa ($cor \approx 0.18$), temperatura del punto de rocío ($cor \approx 0.11$) y presión ($cor \approx 0.10$) y condición de cielo visible. Estas con mes del año y periodo del día fueron las seleccionadas como conjunto de variables predictoras para ambos modelos.

El modelo de clasificación XGBoost ofrece un desempeño equilibrado ante clases desbalanceadas, con alta capacidad para detectar la lluvia entre sus predicciones (89% de las horas en las que llueve) y una alta capacidad de detectar cuando no llueve entre sus predicciones (81% de las horas en las que no llueve). Sin embargo, el modelo acierta solo el 19% cuando pronostica lluvia (entre las horas con y sin lluvia). Sugiriendo la inclusión de nuevas variables que aporten información relevante para reducir los falsos positivos (1964). Por otro lado, el modelo predice casi perfectamente los momentos en los que no llueve, siendo el 99% (entre las horas con y sin lluvia). Debido a la alta cantidad de falsos positivos, se pueden generar falsas alarmas de lluvia, lo cual puede impactar en aplicaciones de riego y alerta temprana.

En cuanto al modelo de regresión, este capta la estructura en los momentos de precipitaciones correctamente en la mayoría de ocasiones, pero falla al cuantificar volúmenes elevados de agua caída, reflejando un sesgo de subestimación. El bajo R^2 0.1161 y el relativamente alto RMSE 0.8903 mm/h indican que faltan variables explicativas de procesos extremos (lo cual puede sugerir variables como anomalías)

El modelo de clasificación captura correctamente (89% de las veces en las que llueve) las horas de lluvia. Significando en un avance y el modelo óptimo para no perder casos de lluvia (los cuales son pocos). Sin embargo, al clasificar lluvia solo estaría acertando el 19% de los casos con y sin lluvia. Si bien esta métrica es aún bastante necesaria su mejora para la aplicación sobre alertas tempranas, el modelo si logra clasificar no lluvia (99% de las horas con y sin lluvia) correctamente. Y capturando el 81% de los casos de no lluvia. Por lo cual, a pesar de aún requerir mejoras para clasificar e incurrir en menos errores al clasificar lluvia. El modelo muestra un sólido rendimiento en la clasificación de no lluvia. Donde, el sesgo se presenta principalmente por clases desbalanceadas (lluvia vs. no lluvia), por lo cual la reducción de los falsos positivos (clasificar lluvia cuando realmente no la hubo) mejoraría drásticamente el modelo.

Disponer de clasificaciones horarias confiables (en este caso para la no lluvia) puede fortalecer la toma de decisiones en Rapa Nui, a pesar de no conseguir una buena clasificación para la lluvia. Este modelo, incluso sin incluir mejoras, puede llevar a la reducción de perdidas por sequía (viendo clasificaciones de no lluvia).

6.3 Recomendaciones, limitaciones y líneas de trabajo futuro

Las recomendaciones comienzan por el enriquecimiento del conjunto de variables predictoras. Incluyendo índices de inestabilidad, datos satelitales de densidad de nubes o variables de viento en capas superiores. Siguiendo con la reevaluación del umbral para clasificación de lluvia (mayor a 0.4 mm/h) según patrones estacionales.

Las limitaciones y posibles sesgos de la aplicación vista van desde:

- Imputación de valores faltantes: KNNImputer puede suavizar momentos extremos si los vecinos no representan correctamente los patrones de lluvia intensa. El uso de KNN para la imputación y estimación de valores faltantes se presenta como posible sesgo debido al agrupamiento o estimación hacia muestras con mayor frecuencia en ciertos valores (en caso de agua_caída gran parte en 0 mm/h).
- Subestimación: Se sugiere la falta de variables clave, esta falta puede estar conduciendo a subestimaciones de las magnitudes de precipitación.
- Desbalance: La gran proporción de horas sin lluvia dificulta el entrenamiento del modelo de clasificación y regresión.
- Usar un umbral único (0,4 mm/h) puede ignorar variaciones estacionales en el punto en que una llovizna deja de ser relevante, lo que podría introducir sesgo en diferentes épocas del año.

Las líneas a seguir en un futuro son:

- Ampliar el conjunto de datos: Integrar variables de fuentes satelitales o estaciones cercanas para entregar al modelo una dimensión espacial y comparar patrones.
- Incorporación automática: Agregar nuevos datos de manera automática para adoptar modelos a las nuevas observaciones.
- Modelos estacionales: Realizar versiones del modelo específicas para cada estación del año, aprovechando diferencias climáticas y patrones propios de cada estación.

Chapter 7

Referencias

[1] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). Association for Computing Machinery. https://doi.org/10.1145/2939672.2939785

[2] Syah, M. S. I., Nardi, & Rachmawardani, A. (2025). Evaluation of the XGBoost Model for Rainfall Prediction and Classification Using BMKG Data and OpenWeather API. *Journal of Computation Physics and Earth Science*, 5(1), 21–30. https://doi.org/10.63581/JoCPES.v5i1.03

[3] Kumar, G. D., Tyagi, S., Pradhan, K. C., & Shah, A. (2025). District-Level Rainfall and Cloudburst Prediction Using XGBoost: A Machine Learning Approach for Early Warning Systems. *Informatica*, 49(2), 375–396. https://doi.org/10.31449/inf.v49i2.7612

[4] American Meteorological Society. (s.f.). Drizzle. En *Glossary of Meteorology*. Recuperado el 19 de junio de 2025, de https://glossary.ametsoc.org/wiki/Drizzle

[5] Gultepe, I. (2008). Measurements of light rain, drizzle and heavy fog. En S. Michaelides (Ed.), *Precipitation: Advances in Measurement, Estimation and Prediction* (pp. 59–82). Springer. https://doi.org/10.1007/978-3-540-77655-0_3

[6] International Civil Aviation Organization & World Meteorological Organization. (2021). Considerations on precipitation intensity thresholds (AMOFSG/10-IP/6). Recuperado de https://www.icao.int/safety/meteorology/amofsg/amofsg%20meeting%20material/amofsg.10.ip.006.5.en.pdf

[7] World Meteorological Organization. (s.f.). Drizzle. En *WMO Cloud Atlas*. Recuperado el 19 de junio de 2025, de https://cloudatlas.wmo.int/es/drizzle.html

[8] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

[9] Chen, T. (2025). XGBoost Documentation. Recuperado el 24 de junio de 2025, de https://xgboost.readthedocs.io/en/latest/

[10] Mujahid, A., & Rafique, M. (2024). *Comparative analysis of missing value imputation techniques for time series data*. Journal of Data Analysis and Information Processing, 12(2), 151–172. https://doi.org/10.4236/jdaip.2024.122009

[11] López-Fernández, J. J., García-Torres, M., Luna, J. M., Ventura, S., & Riquelme, J. C. (2024). *Evaluating missing data imputation techniques in multivariate time series with attention mechanisms*. Pattern Analysis and Applications. https://doi.org/10.1007/s10044-024-01262-3

[12] Chen, T. (2024). *XGBoost model tutorial*. XGBoost Documentation. Recuperado de https://xgboost.readthedocs.io/en/stable/tutorials/model.html

[13] Chen, T., & Guestrin, C. (2016). *XGBoost: A scalable tree boosting system* [Preprint]. arXiv. https://arxiv.org/abs/1603.02754