Universidad Nacional Autónoma de México

facultad de ingeniería

Análisis de algoritmos de machine learning para predicción de aceleraciones sísmicas máximas en México

T E S I S

que para obtener el título de

Ingeniero Geofísico

p r e s e n t a

Jesús Ochoa Contreras

Director de tesis

Dr. Roberto Giovanni Ramírez Chavarría

Ciudad Universitaria, Cd. Mx., 2024

Agradecimientos

A mis padres, que con su amor, ejemplo y apoyo incondicional
me han acompañado en cada paso que doy.

A Nabile, Samara, Erick, Edgar
y todos los compañeros que no menciono por falta de memoria,
por su amistad y apoyo para hacer la carrera más llevadera.

A mis sinodales y profesores de la Facultad de Ingeniería,
por su dedicación en mi enseñanza, tanto académica como profesional y personal.

Al Dr. Roberto Giovanni Ramírez Chavarría,
por su dirección, motivación, consejos y sugerencias durante la realización de la tesis.

Esta tesis fue realizada gracias al Programa de Apoyo a Proyectos para Innovar y Mejorar la Educación (PAPIME) a través del proyecto UNAM-DGAPA-PAPIME PE101524.

Índice general

Índice de figuras

Resumen

Este trabajo tiene como objetivo realizar un análisis de algoritmos de aprendizaje automático para determinar cuál modelo explica mejor el comportamiento de los datos de aceleración sísmica en México. Se detallan los procesos de recolección de datos provenientes de las bases del Instituto de Ingeniería UNAM y del Centro de Instrumentación y Registro Sísmico, así como su procesamiento previo. Se implementan diversos algoritmos, incluidos métodos lineales, reducción de dimensionalidad, árboles de decisión y métodos compuestos. Los resultados se evalúan utilizando el coeficiente de determinación (R^2) y el error cuadrático medio (RMSE), encontrando que el algoritmo de gradiente extremo (XGBoost) produce las mejores predicciones, con una precisión del 75.8 %. Se presenta un ejemplo en el que se comparan los valores obtenidos por el modelo de predicción con los valores reales, demostrando su alta capacidad para adaptarse a la complejidad de los datos sísmicos analizados. Además, se ilustra su aplicación en la simulación de la respuesta sísmica del país ante dos eventos hipotéticos de magnitudes 6.8 y 7.8 en la escala de momento, ambos con epicentro en la trinchera mesoamericana. Se sugiere la posible aplicación de este algoritmo para mejorar la predicción de aceleraciones sísmicas, con el objetivo de optimizar la preparación de los servicios de protección civil ante estos eventos, aunque se reconocen sus limitaciones debido a la falta de datos de calidad para el entrenamiento.

Abstract

This work aims to perform an analysis of machine learning algorithms to determine which model best explains the behavior of seismic acceleration data in Mexico. The data collection processes from the UNAM Engineering Institute and the Seismic Instrumentation and Recording Center databases are detailed, as well as their pre-processing. Various algorithms are implemented, including linear methods, dimensionality reduction, decision trees, and composite methods. The results are evaluated using the coefficient of determination (R^2) and the root mean square error (RMSE), finding that the extreme gradient algorithm (XGBoost) produces the best predictions, with an accuracy of 75.8 %. An example is presented in which the values obtained by the prediction model are compared with the real values, demonstrating its high capacity to adapt to the complexity of the seismic data analyzed. Furthermore, its application is illustrated in the simulation of the country's seismic response to two hypothetical events of magnitudes 6.8 and 7.8 on the moment scale, both with an epicenter in the Mesoamerican trench. The possible application of this algorithm to improve the prediction of seismic accelerations is suggested, with the aim of optimizing the preparation of civil protection services for these events, although its limitations are recognized due to the lack of quality data for training.

Capítulo 1

Introducción

Este capítulo tiene como objetivo proporcionar un marco histórico general sobre la sismicidad en México y sus grandes afectaciones a la sociedad, tanto en términos materiales como humanos. Se ejemplifica la importancia de contar con una instrumentación sísmica adecuada para alertar oportunamente a la población en caso de un sismo de magnitud relevante. A continuación, se presenta la contribución de esta tesis mediante un nuevo método basado en algoritmos de machine learning para estimar aceleraciones sísmicas a partir de las características del evento y las propiedades físicas del suelo. Finalmente, se definen los objetivos del trabajo y se detalla la estructura de la tesis, explicando el contenido y propósito de cada capítulo dentro del proyecto.

1.1. Antecedentes y motivación

Al encontrarse es una de las zonas sismogenéticas más activas del mundo, la sismicidad en México ha sido un tema de gran interés e investigación durante toda su historia, pues representa un riesgo latente en la población en zonas vulnerables, sobre todo aquellas cercanas a la zona de subducción en la trinchera mesoamericana, donde la placa de Cocos se subduce bajo la placa Norteamericana, ocasionando una gran incidencia de sismos de gran magnitud que pueden generar pérdidas materiales millonarias o humanas invaluables.

México ha sufrido numerosos sismos a lo largo de su historia que han dejado huella en la población. Por ejemplo, un sismo el 28 de julio de 1957, de magnitud 7.8, dejando un saldo de 160 muertos y 2500 heridos, así como grandes daños estructurales en la Ciudad de México que ascienden a los 25 millones de dólares. De los daños estructurales ocasionados por este sismo, resalta la caída de la escultura sobre el Ángel de la Independencia, razón

por la cual recibe el nombre de "sismo del Ángel" (CENAPRED, 2024a).

Otro evento importante en la historia de México fue el infame sismo del 19 de septiembre de 1985 ocurrido en Michoacán, con magnitud 8.1 M_W, que dejó a su paso alrededor de 3000 muertes, con estimaciones de hasta 20 mil (SIAP, 2024); el sismo en Chiapas del 8 de septiembre de 2017 con magnitud 8.2 cuyas afectaciones se vieron acentuadas por un sismo magnitud 7.1 con epicentro en Puebla apenas 12 días después, dejando detrás una cifra de fallecidos que asciende a los 400 personas (CENAPRED, 2024b) y costos materiales de 62 mil millones de pesos (García Arróliga et al., 2019).

Todos estos eventos comparten la tragedia de que la mayor parte de las pérdidas materiales y humanas sucedieron en la Ciudad de México, ocasionado debido a la alta densidad poblacional y las condiciones del suelo lacustre sobre el cual está construida, lo que amplifica las ondas sísmicas a través del efecto de sitio.

1.2. Planteamiento del problema

México se encuentra en una posición desafortunada para la población que sufre los estragos de los sismos, pero también tiene el privilegio de poder contar con registros de aceleración de calidad asociados a sismos de gran magnitud, por lo que su análisis para fines de investigación ha traído consigo nuevo entendimiento y numerosos trabajos acerca de la mecánica de los sismos en límites de placa convergentes.

Además, la cercanía a una zona sísmica tan activa ha requerido la toma de acciones para la prevención de desastres, como la colocación de sensores que puedan alertar a la población de manera temprana ante la ocurrencia de un sismo. Estos sensores deben estar colocados en lugares estratégicos según su riesgo de ocurrencia de un sismo, de forma que puedan registrar si un movimiento es potencialmente peligroso de manera oportuna.

La tarea de colocación de estaciones sísmicas con fines de investigación o prevención está bajo la responsabilidad de varias dependencias, como el Servicio Sismológico Nacional, el Instituto de Ingeniería UNAM, el Centro de Instrumentación y Registro Sísmico de la Ciudad de México, entre otras. La información recopilada a lo largo de los años permite tener estimaciones acerca de la susceptibilidad de un determinado sitio ante un sismo, utilizando datos conocidos para calcular el movimiento esperado para un sismo dado. Aunque estas estimaciones son precisas a través de modelos matemáticos o numéricos que describen su comportamiento, son calculadas para cada sitio de manera particular, por lo que puede volverse impráctico contar con estimaciones para todo el país.

1.3. Contribución

Para resolver la dificultad de estimación de susceptibilidad por riesgos sísmicos de manera precisa, se propone un método que utiliza algoritmos de machine learning para generar datos de aceleración máxima ante un sismo hipotético. Este modelo utiliza datos de aceleración generados a lo largo de la historia para sismos de distintas características y magnitudes, asociando la respuesta de cada sitio con base en sus propiedades elásticas y geológicas. Este método debe ser entrenado con una gran cantidad de datos de calidad para poder generar predicciones precisas, por lo que la recopilación de datos es la principal tarea y posible dificultad de su implementación. Sin embargo, la gran ventaja es que evita la necesidad de estimar leyes de atenuación para cada sitio individualmente, y puede adaptarse al comportamiento de cualquier sitio en el país.

1.4. Objetivos

Realizar una investigación exploratoria de distintos algoritmos de machine learning con aplicación en la predicción de aceleraciones sísmicas máximas en México. Asimismo, evaluar el desempeño de los algoritmos implementados para determinar las ventajas y desventajas de éstos en términos de su exactitud, precisión, y eficiencia computacional.

1.5. Estructura de la tesis

La tesis está dividida en 5 capítulos, donde el primero es el presente, que proporciona el contexto y la justificación del estudio. El segundo capítulo revisa el estado del arte en la predicción de aceleraciones sísmicas máximas, ya sea utilizando machine learning u otras técnicas matemáticas, así como la teoría básica de los algoritmos utilizados en el análisis. El tercer capítulo describe la base de datos para el entrenamiento, revelando información básica acerca de los datos mediante gráficos y análisis estadísticos. El cuarto capítulo contiene la metodología utilizada para la recopilación y preparación de los datos, así como la implementación de los algoritmos de machine learning. El quinto capítulo presenta los resultados y su análisis en términos de precisión y eficiencia, así como una demostración de la simulación de sismos y estimación de aceleraciones máximas. Finalmente, el sexto capítulo concluye la tesis con una discusión de los resultados y recomendaciones para futuras investigaciones.

Este primer capítulo ha establecido el contexto y la importancia de estudiar la sismicidad en México, así como la necesidad de contar con técnicas avanzadas para la predicción de aceleraciones sísmicas. Se han revisado los conceptos teóricos y los métodos tradiciona-

les utilizados en la sismología, destacando sus limitaciones y la motivación para explorar enfoques basados en machine learning. En el siguiente capítulo, se profundizará en las generalidades teóricas de este trabajo, describiendo las matemáticas detrás de los algoritmos de machine learning seleccionados y su implementación para mejorar la precisión en la estimación de aceleraciones sísmicas. Esta transición permitirá comprender mejor cómo los conceptos teóricos se aplican en la práctica para abordar los desafíos identificados en este capítulo inicial.

El código implementado, así como ejemplos de uso, pueden ser consultados en el siguiente enlace: https://github.com/JOchoa51/tesis_geofisica

Capítulo 2

Generalidades

Este capítulo abarca la información teórica fundamental necesaria para contextualizar el presente trabajo. Se exploran temas como la sismicidad en México y su origen, la sismicidad histórica, los métodos de registro de eventos sísmicos y sus impactos en la sociedad. Además, se proporciona información sobre las técnicas actuales para la predicción de aceleraciones sísmicas, como los mapas de zonificación y las leyes de atenuación matemáticas, destacando sus ventajas y desventajas. Asimismo, se introduce la teoría detrás de los algoritmos utilizados en este estudio y se compara con las técnicas actualmente empleadas para la estimación de intensidades sísmicas

2.1. Sismicidad en México

La sismicidad en la República Mexicana es ocasionada principalmente por un régimen compresivo de la placa de Cocos, Rivera y Norteamericana, donde las dos primeras se subducen bajo la última (ver Figura 2.1). Este margen de placas está asociado a la sismicidad del Anillo de Fuego del Pacífico, una zona alrededor del Océano Pacífico con muy alta sismicidad y actividad volcánica, responsable de algunos de los sismos de mayor magnitud en la historia (USGS, 2009).

En México, múltiples sismos de gran magnitud han azotado el territorio, en ocasiones trayendo consigo destrucciones y pérdidas humanas incalculables. Sin duda, el mejor ejemplo de un evento de este tipo ese el sismo del 19 de septiembre de 1985, de magnitud 8.1 con epicentro en las costas del estado de Michoacán. Este sismo generó devastación a lo largo de las regiones circundantes, pero se vio especialmente acentuado en la Ciudad de México, debido a la naturaleza del suelo en el que está construida. Múltiples edificios co-

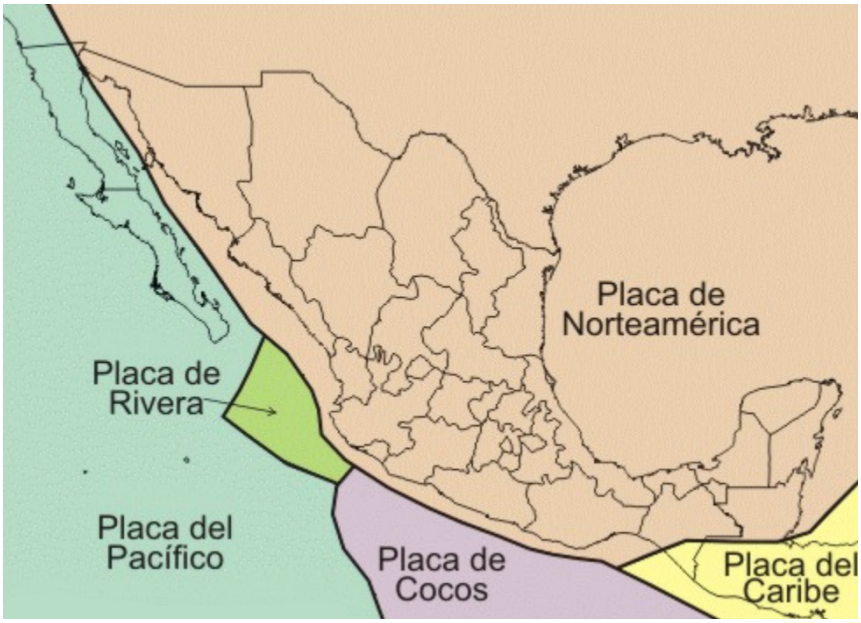

Figura 2.1: Esquema de las placas tectónicas que conforman el territorio mexicano

lapsaron ante este sismo, generando incontables pérdidas materiales y humanas, solamente para ser testigos de una réplica con magnitud 7.6 que terminó de diezmar las estructuras reblandecidas por el evento principal.

Este sismo fue el primer sismo de gran magnitud en ser registrado por la Red Acelerográfica del Instituto de Ingeniería UNAM recientemente colocados en la zona como parte de una colaboración con el gobierno japonés, generando así un gran acervo de registros sísmicos tanto en zonas muy cercanas al epicentro como en estaciones tan alejadas como 400 km (IdeI UNAM, 2017). Este sismo supuso un parteaguas en la importancia de una adecuada instrumentación sísmica en el país.

Otros eventos que se han beneficiado de la posterior colocación y mejora de instrumentos sísmicos para la detección temprana son los sismos del 7 de septiembre de 2017 de magnitud 8.2 en Tehuantepec, cuya alerta temprana gracias a la instrumentación ayudó a salvar vidas; el sismo de Michoacán el 19 de septiembre de 2022 con magnitud 7.7, sucedido solamente unos minutos después del simulacro nacional conmemorativo del sismo de 1985 exactamente 37 años antes.

Un evento que vive en la memoria de los mexicanos y que nos recuerda de la naturaleza imprevisible de los sismos es el ocurrido el 19 de septiembre de 2017, de magnitud 7.1 en el estado de Puebla. El epicentro de este sismo era demasiado cercano a la zona del Valle de México para ser detectado por los sismómetros responsables de emitir la alerta sísmica (Fariza, 2017), por lo que tomó por sorpresa a los habitantes de las zonas cercanas.

El movimiento dejó numerosos daños en los estados de Puebla, Morelos y la Ciudad de México, provocando el colapso de más de 44 mil estructuras y la pérdida de más de 300 vidas, dejando más de 6000 heridos (Gobierno de México, 2024).

Aunque es imposible no hacer una correlación inconsciente entre estos eventos, dado que todos ocurrieron durante el mes de septiembre, debe recordarse que no existe evidencia geológica que respalde la correlación entre la ocurrencia de un sismo en una determinada fecha, pues solamente se trata de coincidencias (Staff, 2022).

Debido a la zona de alta susceptibilidad sísmica en la que se encuentra México, existe un riesgo latente de eventos de gran magnitud que pueden impactar en cualquier momento, por lo que contar con una instrumentación sísmica confiable que permita mitigar en la medida de lo posible las afectaciones por este tipo de desastres es de vital importancia.

2.2. Instrumentación sísmica

La instrumentación sísmica permanente en México está a cargo de diversas instituciones y dependencias, cada una cubriendo un área distinta y cuyos sensores son de distintas características técnicas. De manera general, la mayoría de los sensores destinados al registro de sismos son operados por el Servicio Sismológico Nacional (SSN), que cubre gran parte del país con su red de 62 sismógrafos de banda ancha (SSN, 2015); el Instituto de Ingeniería de la UNAM que cuenta con una red acelerográfica de 90 sensores en la zona sur del país y una mayor densidad de estaciones en el Valle de México (IdeI UNAM, 2017); y el Centro de Instrumentación y Registro Sísmico de la Ciudad de México (CIRES), operando 80 estaciones sísmicas en el Valle de México para registro y mapeo de características sísmicas en la capital del país (CIRES, 2024), así como alertamiento temprano en la zona sur de México.

Para fines de alerta ante sismos, el Sistema de Alerta Sísmica Mexicano (SASMEX) cuenta con sensores colocados de manera estratégica a lo largo de la costa del Pacífico en México, así como cobertura total en los estados de Guerrero, Oaxaca y Puebla. Este sistema permite alertar hasta con decenas de segundos ante las llegadas de las ondas sísmicas a las regiones en riesgo (CIRES, 2024).

Si bien los sismógrafos son muy útiles para fines de investigación, juegan un papel muy importante en la detección temprana de sismos para dar alerta a la población. Cabe destacar que el Servicio Sismológico Nacional no opera la alerta sísmica de la Ciudad de México, pues esa tarea está a cargo del SASMEX.

Figura 2.2: Zonificación sísmica en México para un periodo de 400-500 años (Esteva, 1970).

2.3. Análisis de datos para predicción de aceleraciones sísmicas

Los métodos actuales para estimar las aceleraciones sísmicas en México contemplan métodos de modelación numérica y matemática, así como estimaciones realizadas a través del análisis de la sismicidad histórica con fines ingenieriles, los cuales son utilizados para el diseño de estructuras susceptibles a esfuerzos sísmicos. Aunque los mapas de zonificación sísmica son una buena estimación de la intensidad esperada, solamente ofrecen una visión general del riesgo en una zona y no contienen datos precisos de la respuesta sísmica ante un determinado evento, pues en ocasiones suponen condiciones ideales que pueden no representar a todas las zonas por igual. Tal es el caso de la regionalización sísmica de México para fines de ingeniería de Esteva (1970), que presenta una regionalización en todo el país suponiendo que el suelo está compuesto por conglomerados compactos en su totalidad, posiblemente con fines de simplificar su trabajo (Figura 2.2).

Si se desea conocer el tipo de respuesta que tiene un sitio en particular se debe realizar un análisis de la ley de atenuación sísmica que le aplique, a través de modelaciones matemáticas definidas a partir del movimiento del terreno y el mecanismo de falla asociado

al sismo en cuestión. La relación del movimiento del suelo en su forma más sencilla se determina a partir de varios parámetros según Bozorgnia y Bertero (2004):

$$\ln Y = C_1 + C_2 M_W - C_3 \ln R - C_4 R + \varepsilon \tag{2.1}$$

donde el valor de Y es el movimiento del terreno asociado a un sismo con magnitud M_W y a una distancia R, con un término de error ε y la influencia de los parámetros C_1, C_2, C_3 y C_4, que describen el movimiento fuerte del suelo, la atenuación geométrica y anelástica de las ondas ocasionadas por el amortiguamiento en distintos materiales. Obsérvese el término $C_3 \ln R$, afectado por un factor logarítmico, que indica una disminución exponencial en el movimiento del suelo con una mayor distancia al epicentro. Aunque obtener leyes de atenuación sísmica es un método muy preciso, debe ser calculado de manera independiente para cada sitio de manera particular con datos sísmicos de calidad (Flores, 2015), por lo que puede tornarse impráctico.

2.4. Algoritmos de *machine learning* en geofísica

Debido a que la geofísica implica intrínsecamente el análisis de grandes cantidades de datos y relaciones matemáticas complejas para su modelado e inversión, el advenimiento de los algoritmos computacionales en lo años recientes han supuesto un gran salto en la eficiencia de estos análisis, tanto en el tiempo de procesamiento como en la calidad de los resultados.

Algunos de los algoritmos utilizados en la geofísica, sobre todo en problemas que requieren la optimización de modelos, incluyen métodos de gradiente, algoritmos genéticos, recristalización simulada, enjambres de partículas, redes neuronales, entre otros (Qadrouh et al., 2019). Así pues, el objetivo de la inteligencia artificial en general es simular el comportamiento inteligente de los humanos (Samuel, 1959), concepto que fue concebido a mediados del siglo XX. El objetivo principal del machine learning es la predicción del comportamiento de una variable dado un conjunto de datos de entrada a través de algoritmos que serían extremadamente complicados de manera usual, o en modelos donde el uso de técnicas convencionales no ofrecen los resultados esperados.

Por ejemplo, algunos de los trabajos más relevantes en geofísica incluyen: la utilización de redes neuronales auto-supervisadas para la supresión del ruido aleatorio en señales sísmicas (Birnie et al., 2021), la obtención de soluciones de onda restringidas por la ecuación de Helmoltz a partir de funciones aprendidas mediante machine learning (Alkhalifah

et al., 2021), identificación rápida de reservorios marinos de gas de alta calidad utilizando bosques aleatorios (Zhu et al., 2021), predicción de la concentración de elementos traza en rocas (Zhang et al., 2021), mejora en la predicción de litofacies a partir de datos de rayos gamma en pozos con datos limitados (Wood, 2021), uso de redes neuronales para la expansión en el etiquetado automático de datos sísmicos (Li et al., 2020), estimaciones rápidas de magnitud en los sismos usando redes neuronales convolucionales (Meng et al., 2023), predicción de fallas sísmicas en condiciones de laboratorio a partir del análisis de señales previas (Rouet-Leduc et al., 2017), detección automática de anomalías geológicas superficiales a partir de imágenes multiespectrales (Nwaila et al., 2022), entre otros.

Entre los trabajos previos utilizando machine learning en geofísica, destaca el trabajo de Joshi et al. (2024), donde se diseña un algoritmo basado en modelos híbridos, basados en kernels, árboles de decisión y regresión simple, con el objetivo de predecir el valor de PGA en sismos de Japón. Este algoritmo demuestra tener un error cuadrado medio de la mitad de los métodos convencionales, siendo probado para sismos en Irán y demostrando una capacidad de predicción más precisa que las leyes de atenuación locales. El trabajo de Joshi et al. (2024) demuestra que la utilización de algoritmos de machine learning para el fin que en esta tesis se propone es perfectamente viable, siempre y cuando los datos de entrenamiento sean vastos y de calidad.

Aunque el uso de algoritmos de machine learning en geofísica presenta la ventaja de no requerir el mismo conocimiento profundo de ecuaciones o teoría física para crear modelos matemáticos por métodos convencionales, es esencial que el programador cuente con los conocimientos necesarios para garantizar la creación y validación de un modelo confiable. Además, una de las principales desventajas es que la recopilación de suficientes datos para entrenar el modelo puede representar un gran desafío. Sin embargo, cuando los algoritmos están bien entrenados, demuestran una capacidad de adaptación y resolución excepcional (Kim & Nakata, 2018).

En el ámbito de las geociencias en general, ha tomado lugar un gran número de avances en la aplicación de algoritmos de machine learning, abarcando una cantidad formidable de algoritmos, por ejemplo: árboles de decisión, bosques aleatorios, máquinas de soporte vectorial, procesos Gaussianos, redes neuronales profundas, redes neuronales convolucionales y redes generativas adversarias, por nombrar algunos.

2.5. Generalidades de los algoritmos utilizados

Esta sección tiene como objetivo presentar la teoría matemática básica detrás de los algoritmos seleccionados para su análisis en este trabajo, centrándose en algunos de los más comunes, como los modelos lineales, los árboles de decisión y los modelos de gradiente. Las redes neuronales no se incluyen en el análisis, ya que estos algoritmos pertenecen a un área de investigación distinta que queda fuera de los objetivos planteados.

2.5.1. Regresión lineal

La regresión lineal representa el método de relación entre variables más básico, donde la variable objetivo puede tener una o varias variables explicativas. En general, el modelo consta en adaptar una línea recta a las observaciones $(x_1, y_1), \ldots, (x_n, y_n)$ (Kroese et al., 2019). Debido a que este método puede tener múltiples variables, es mejor expresar el modelo matemático en términos matriciales:

$$y = m\mathbf{X} + b \tag{2.2}$$

donde m representa la matriz de parámetros de entrada, los cuales controlan el valor de la variable objetivo y con los coeficientes almacenados en la matriz $\mathbf{X}$ y su intersección con el eje objetivo, b, de tal manera que los valores residuales entre la línea recta y los puntos de datos sea mínimo. Este modelo se resuelve utilizando mínimos cuadrados, pues la función de costo $L(\varphi)$ que minimiza es la diferencia entre los valores estimados y la función real (Pedregosa et al., 2011):

$$L(\varphi) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - y)^2 \tag{2.3}$$

donde n es el número de muestras en los datos, y_i es el valor estimado y y es el valor real.

La forma general para la solución de un sistema determinado descrito por la matriz de parámetros $\mathbf{m}$, los coeficientes $\mathbf{G}$ y el objetivo $\mathbf{y}$ como $\mathbf{y} = \mathbf{Gm}$ es:

$$\mathbf{m} = (\mathbf{G}^T\mathbf{G})^{-1}(\mathbf{G}^T\mathbf{y}) \tag{2.4}$$

La regresión lineal por mínimos cuadrados puede ser altamente susceptible a colinealidad, por lo cual se desarrollaron métodos que apliquen un factor de regularización a

los coeficientes estimados con el objetivo de evitar el sobreajuste a los datos.

El problema de la colinealidad (o multicolinealidad cuando se trata con más de dos variables) surge cuando dos o más variables predictoras tienen una relación lineal entre ellas, potencialmente afectando el desempeño de la predicción. El método de regresión de cresta ofrece un ajuste lineal con un factor de regularización L2 (cuadrático) aplicado a los coeficientes m_i del modelo, cuya influencia en el ajuste del modelo está controlado por el parámetro λ:

$$L(\varphi) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - y)^2 + \lambda\sum_{i=1}^{n} m_i^2 \tag{2.5}$$

Este modelo, llamado regresión lineal con regularización L2 o regresión lineal de cresta, garantiza una mayor penalización a los coeficientes con mayor magnitud (IBM, 2023).

Aunque la regularización L2 en ocasiones se presenta como una alternativa robusta, pueden existir casos en los que un factor cuadrático de regularización es demasiado pesado y una regularización lineal es insignificante. Debido a este problema surgen los métodos con combinación de regularizaciones L1 y L2, de tal manera que los coeficientes no sean penalizados con tanta dureza con L2 ni eliminados totalmente con L1, aplicando una regularización intermedia en su lugar. Este método es llamado malla elástica, la cual puede penalizar todos los coeficientes colineales, a diferencia de la penalización por L1 y L2 solamente (Pedregosa et al., 2011), prometiendo generar resultados con mayor precisión al eliminar coeficientes poco relevantes (Zou & Hastie, 2005). Su función de costo está definida por la fuerza de la regularización, λ, y la proporción de regularizaciones L1/L2, α:

$$L(\varphi) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - y)^2 + \lambda\alpha\sum_{i=1}^{n} m_i^2 + \frac{\lambda(1-\alpha)}{2}\sum_{i=1}^{n} m_i \tag{2.6}$$

2.5.2. Árboles de decisión

Los árboles de decisión son una familia de algoritmos de aprendizaje no paramétrico (es decir, no dependen de ninguna función matemática), cuyas predicciones son basadas en estructuras similares a árboles, donde los datos se dividen según los criterios establecidos (Kristori, 2024). Una gran ventaja de este tipo de algoritmos es que, cuando son aplicados de forma independiente, son muy intuitivos y pueden ser visualizados fácilmente.

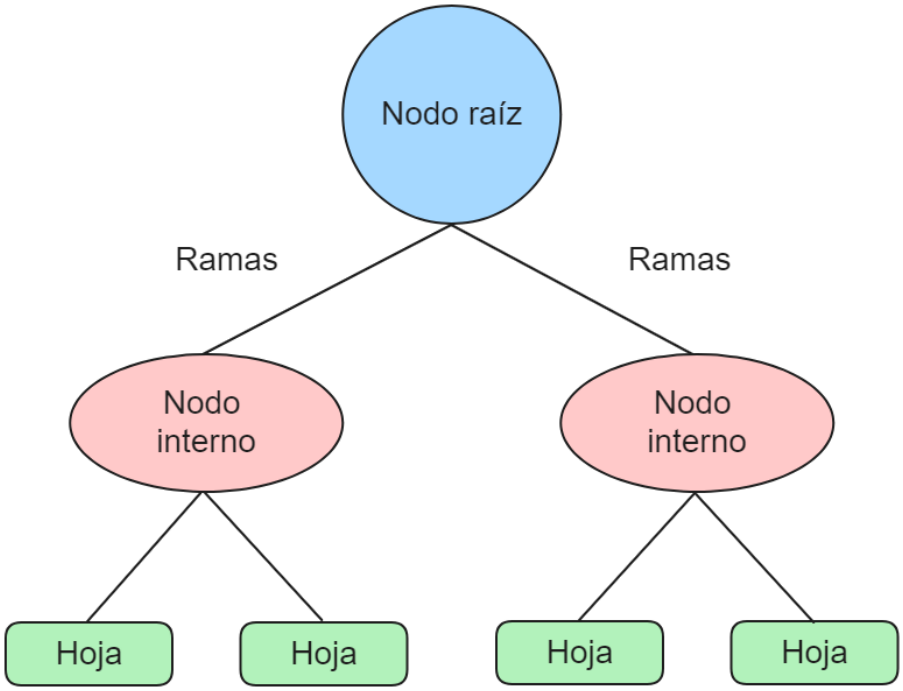

Figura 2.3: Estructura de un árbol de decisión simple. Modificado de Kroese et al. (2019).

En la Figura 2.3 muestra la estructura general de un árbol de decisión, donde las predicciones se encuentran en las hojas, generadas a partir de divisiones de los datos basadas en reglas inferidas por el algoritmo. A medida que aumenta la profundidad del árbol, es decir, a mayor número de divisiones antes de llegar a una hoja, las reglas de decisión se vuelven más complejas, lo que puede generar predicciones más precisas, pero también existe el riesgo de sobreajuste si hay muchas características y pocas observaciones (Pedregosa et al., 2011). La función matemática optimizable de un árbol de decisión es la siguiente:

$$L(\varphi) = \sum_{i=1}^{n} l(y, y_i) + \sum_{i=1}^{n} \Omega(f_i) \tag{2.7}$$

donde l es una función diferenciable que mide la diferencia entre la variable objetivo y y el valor predicho y_i. El término $\Omega(f_i)$ es un factor de penalización para evitar el sobreajuste en los árboles (Natekin & Knoll, 2013).

2.5.3. Métodos de conjunto

Aunque los árboles de decisión simples pueden ser muy buenos estimadores por sí solos, pueden llegar a ser insuficientes dependiendo de la complejidad de los datos con los

que se trabaje. Debido a esto, existen métodos que generan resultados a partir del uso de múltiples predictores simples, usualmente árboles de decisión (Natekin & Knoll, 2013), cada uno de los cuales es entrenado en un subconjunto de datos diferente y aleatorio del set de datos completo. Este grupo de predictores son evaluados en conjunto para optimizar una función de costo diferenciable arbitraria, aumentando así su precisión y generalizabilidad (Pedregosa et al., 2011).

Los árboles de decisión potenciados por gradiente emergen como una solución para optimizar funciones más complejas que las de un árbol de decisión simple, permitiendo encontrar relaciones más complejas y generar predicciones con mayor precisión. Estos algoritmos se optimizan de manera iterativa hasta alcanzar un nivel de tolerancia especificado, creando una cantidad determinada de árboles en cada iteración, y donde la predicción final se obtiene a partir del promedio de las predicciones de todos los árboles creados durante la optimización.

La función de optimización, que parte de la Ecuación 2.7, es la siguiente:

$$L(\varphi)^{(t)} = \sum_{i=1}^{n} l(y, y_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{2.8}$$

donde t representa la t-ésima iteración de optimización, y la función f_t es añadida a la función de costo de tal manera que optimice el modelo en la mejor manera. La precisión del modelo es evaluada calculando su error cuadrado medio (MSE).

Otro tipo de método de conjunto muy utilizado son los bosques aleatorios, que eliminan la necesidad de calcular gradientes y funciones de costo diferenciables, pues solamente generan árboles entrenados en diferentes sets de datos y predicen un valor promedio de todos los árboles creados. Una de las condiciones de este método es que cada árbol, junto con sus parámetros de división y predicciones finales, deben ser independientes entre sí, de manera que las decisiones tomadas reflejen de manera precisa la estructura real de los datos (Kroese et al., 2019).

De forma similar a los árboles de decisión, los bosques aleatorios controlan la calidad de la estimación utilizando hiperparámetros como la máxima profundidad de cada árbol (número de divisiones hechas) y el número mínimo de elementos para considerarse una hoja. La función de optimización de los bosques aleatorios también debe tomar en cuenta el número de árboles que genera, pues de esto dependerá la complejidad del modelo y su desempeño en datos de validación.

Si los árboles independientes creados por bosques aleatorios y árboles potenciados por gradiente no logran captar del todo la estructura en datos muy complejos, entonces es necesario relacionar cada árbol, de manera que exista comunicación de los errores de cada uno para que sean corregidos en el siguiente árbol. Esto garantiza que el error al final de la creación de todos los árboles sea el mínimo posible, con la diferencia de que la decisión final se pondera según el nivel de error de cada árbol, a diferencia de los bosques aleatorios que promedia todos sus árboles con igual peso (Pro, 2016).

Este tipo de algoritmo es llamado gradiente extremo, conocido también por el nombre de su implementación computacional más utilizada, *XGBoost* (Chen & Guestrin, 2016). Los hiperparámetros que se deben tomar en cuenta para la generación de estos árboles son muy similares a los algoritmos basados en el mismo principio, y solamente se añade la tasa de aprendizaje bajo la cual el algoritmo generará nuevos árboles en cada iteración. Al igual que los árboles impulsados por gradiente y los bosques aleatorios, los árboles de gradiente extremo son evaluados calculando el error cuadrado medio de la predicción respecto a la variable real.

2.5.4. Reducción de dimensionalidad

Si bien muchos de los algoritmos existentes están optimizados para trabajar con datos con alta dimensionalidad, por ejemplo los árboles de decisión, muchos otros pueden verse afectados negativamente al tener dificultades para estimar los hiperparámetros óptimos (Rogers & Girolami, 2020).

Debido a esto y para evitar problemas relacionados con la maldición de la dimensionalidad[1], existen algoritmos que reducen las dimensiones de los datos analizando la contribución de cada característica a la varianza. Obteniendo las dimensiones que expliquen la mayor parte de la varianza de los datos se puede diseñar un conjunto de datos de mucho menor tamaño y que aún mantenga la mayor parte de las características de los datos originales (Pedregosa et al., 2011).

Según Rogers y Girolami (2020), el análisis de componentes principales, comúnmente abreviado como PCA, por sus siglas en inglés, supone una matriz de datos $\mathbf{X}$, de la cual se calcula su matriz de covarianza $\mathbf{C}$ después de normalizar y centrar las n observaciones:

$$\mathbf{C} = \frac{1}{n}\mathbf{X}^T\mathbf{X} \tag{2.9}$$

[1]La maldición de la dimensionalidad ocurre cuando existen datos distribuidos en demasiadas dimensiones, lo que hace que los datos tengan una poca probabilidad de compartir características bajo las cuales se puedan agrupar (Verleysen & François, 2005).

Posteriormente se obtiene cada uno de los vectores y valores característicos de $\mathbf{C}$, los cuales se ordenan según su magnitud. En este paso se eligen k vectores característicos en función de la cantidad de dimensiones a las cuales se desea reducir la matriz $\mathbf{X}$. De este modo, la matriz de datos reducidos se calcula como una proyección de $\mathbf{X}$ sobre la matriz de vectores propios $\mathbf{V_k}$:

$$\mathbf{Z} = \mathbf{X_C V_k} \tag{2.10}$$

donde $\mathbf{Z}$ es la matriz de datos reducidos. Por otra parte, el algoritmo PCA puede considerarse un método de extracción de características también, ya que permite manejar datos de alta dimensionalidad utilizando solo las características más relevantes para el análisis (Kroese et al., 2019).

Dado que el método de PCA solamente utiliza la matriz $\mathbf{X}$ para la transformación, no asegura que sea relevante para los valores de $\mathbf{Y}$, es decir, el objetivo. Para solucionar este problema, existen métodos como los mínimos cuadrado parciales (PLS), que realiza una descomposición simultánea de las matrices $\mathbf{X}$ y $\mathbf{Y}$, asegurando que los vectores y valores característicos utilizados durante la proyección expliquen la correlación entre ambas matrices tanto como sea posible (Pedregosa et al., 2011). Además, PLS es particularmente utilizado cuando se requiere predecir un conjunto de variables independientes a partir de un conjunto muy grande de variables dependientes (Abdi, 2003).

Este capítulo ha proporcionado un panorama general de la sismicidad en México, explorando su origen, historia y los métodos empleados para el registro de eventos sísmicos. Se han comentado tanto los efectos materiales como humanos de los sismos, enfatizando la necesidad de una instrumentación sísmica adecuada para la alerta temprana. Por otra parte, se han examinado las técnicas actuales para la predicción de aceleraciones sísmicas, como los mapas de zonificación y las leyes de atenuación, evaluando sus ventajas y desventajas. Finalmente, se ha proporcionado un contexto básico de los algoritmos de machine learning elegidos. Con las bases teóricas y metodológicas establecidas, en el siguiente capítulo se detallará la recopilación, unificación y análisis exploratorio de los datos sísmicos utilizados durante la tesis, que fundamentarán la implementación y evaluación de los algoritmos presentados.

Capítulo 3

Base de Datos

En este capítulo se presenta el procedimiento de análisis estadístico, transformación y unificación del catálogo de sismos recopilado, con el objetivo de asegurar un análisis correcto por los algoritmos de aprendizaje automático. Se detallan los procesos de transformación de escalas de magnitud sísmica, análisis estadístico y descripción de la base de datos, para así comprender de manera general el tipo de datos con los que se está trabajando y poder seleccionar de manera más precisa los tipos de algoritmos que pueden ofrecer mejores resultados.

3.1. Recopilación de datos

El 29 % de los acelerogramas utilizados durante el entrenamiento de los modelos fueron obtenidos de una base de datos del Centro de Instrumentación y Registro Sísmico (CIRES) de la Ciudad de México, la cual fue solicitada de manera directa a la dependencia para los fines que a este proyecto conciernen. Los acelerogramas restantes se distribuyen en 68 % pertenecientes al Instituto de Ingeniería UNAM y 3 % correspondiente a acelerogramas de la Benemérita Universidad Autónoma de Puebla (BUAP), dando un total de 2710 registros. Es importante mencionar que las estaciones del CIRES solamente cubren el área del Valle de México, mientras que los demás acelerogramas abarcan la zona sur del país casi en su totalidad.

Los eventos sísmicos asociados a estos acelerogramas cumplen con ciertos criterios de selección para asegurar su calidad y consistencia, los cuales se enlistan a continuación:

- Todos los sismos se encuentran en el periodo del 05-01-1971 al 14-12-2023

- Salvo sismos locales en el Valle de México, la magnitud de todos los eventos se encuentra por encima de 4.5 en la escala de magnitud de momento.
- Salvo sismos locales en el Valle de México, todos los sismos están asociados a la tectónica de la zona de subducción de la trinchera Mesoamericana o fallas laterales asociadas al contacto de la placa de Cocos y Rivera.
- Para asegurar un nivel de señal/ruido utilizable, la distancia máxima de una estación al epicentro de un sismo es de aproximadamente 1000 km.

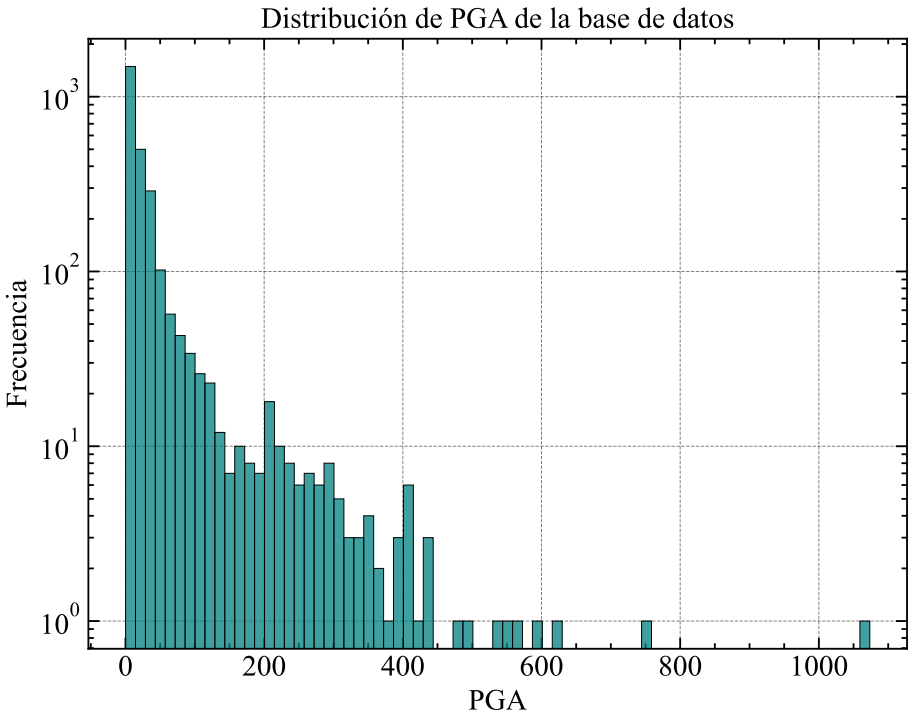

Figura 3.1: Distribución de los valores de PGA en la base de datos utilizada

Dado que los archivos ASA2.0 cuenta con un formato estandarizado y uniforme, es posible diseñar funciones en Python para iterar sobre cada archivo y extraer los datos necesarios, realizando las operaciones matemáticas necesarias de manera simultánea (por ejemplo, el cálculo de espectros de frecuencias). Los parámetros extraídos de los archivos y que son relevantes para el entrenamiento de los modelos se describen a continuación:

- Magnitud del sismo (medido en distintas escalas que posteriormente fueron unificadas)
- Distancia de la estación al epicentro en kilómetros
- Profundidad hipocentral en kilómetros

- Geología/tipo de suelo donde está colocado el sensor

La base de datos completa consta de alrededor de 81,000 elementos, sin tomar en cuenta la señal por sí misma.

3.2. Unificación del catálogo

La base de datos contempla eventos registrados desde el año 1971 hasta 2023, con instrumentos de distintas tecnologías, precisiones, instituciones responsables y parámetros medidos. Si bien el registro de sismos en los últimos años ha mejorado considerablemente, los sismos de hace más de 20 años aún tienen en su registro parámetros de medición que no son usados actualmente, por ejemplo mediciones analógicas que tuvieron que ser digitalizadas manualmente o de forma automática, o escalas de magnitud sísmica obsoletas.

Una gran ventaja de utilizar sismogramas con el formato estándar ASA2.0 es que la distribución de los datos dentro del archivo es consistente, pero deben ser revisados y limpiados cuidadosamente antes de pasar al procesamiento. Uno de los valores con mayor influencia en el análisis y mayor discrepancia entre sí es la escala de magnitud registrada en cada archivo. Aunque para sismos recientes es un estándar utilizar la magnitud de momento (M_W), existen registros con escalas locales y relacionadas a la amplitud de ondas P o S, o asociadas a la duración del registro (magnitud de onda de cuerpo: M_b; magnitud de onda superficial: M_S; magnitud de coda: M_C).

Estas magnitudes pueden no estar bajo la misma escala y no representar la misma energía liberada, por lo que es necesario unificar estas escalas para obtener sus valores correspondientes en la escala M_W. Para lograr esta transformación se utilizan ecuaciones obtenidas de manera empírica a partir de la determinación de coeficientes lineales y cuadráticos mediante regresiones estadísticas para sismos en la República Mexicana utilizando datos del *International Seismological Centre, ISC* y del *United States Geological Survey, USGS* (Sawires et al., 2019). Estas ecuaciones resultan en un valor de M_W^*, que es comparable con los valores de M_W medido directamente, por lo que permiten realizar el análisis correctamente.

$$M_W^* = (5.58 \pm 0.29) - (0.68 \pm 0.10)M_S + (0.13 \pm 0.01)M_S^2 \tag{3.1}$$

$$M_W^* = (-1.36 \pm 0.13) + (1.35 \pm 0.15)M_b \tag{3.2}$$

$$M_W^* = (-0.31 \pm 0.26) + (1.06 \pm 0.21)M_C \tag{3.3}$$

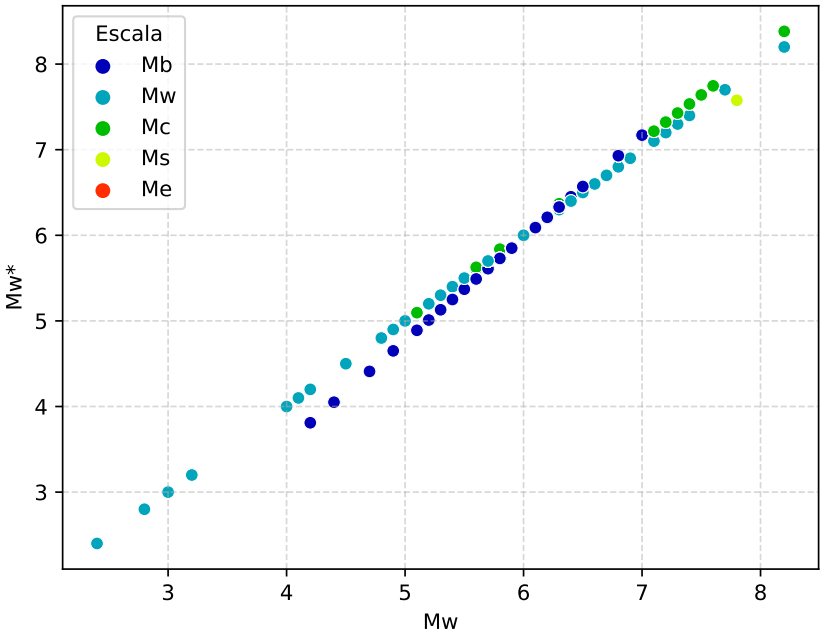

Figura 3.2: Distribución de magnitudes sísmicas en la base de datos después de su transformación. Los puntos indicados como M_W se refieren a la magnitud original registrada en el archivo, mientras que el resto son las magnitudes transformadas a unidades de M_W^*.

Después de la transformación se puede apreciar en la Figura 3.2 que la nueva magnitud equivalente M_W^* sigue de manera muy precisa la tendencia de la magnitud de momento real, por lo que la transformación logra obtener valores de magnitud de momentos de manera fiable, lo que permite realizar un análisis sin sesgos por escala. Esta transformación es de especial relevancia debido a que son relaciones desarrolladas especialmente para el régimen tectónico de México, por lo que su aplicación asegura que los datos se mantienen utilizables.

3.3. Análisis exploratorio

A partir de la recopilación de diversas fuentes, la base de datos cuenta con un total de 2710 registros, asociados a 142 sismos con magnitudes que van desde 2.4 hasta 8.3 en la escala M_W, registrados en 264 estaciones sísmicas. La gran mayoría de los sismos están asociados a la subducción de la placa de Cocos y Rivera bajo la placa Norteamericana, mientras que otros pocos se deben a la subsidencia del suelo en la Ciudad de México (aquellos con magnitud de 3 o menor y epicentro en el Valle de México). En la Figura 3.3 se presenta la distribución de los epicentros de los sismos analizados, donde se puede

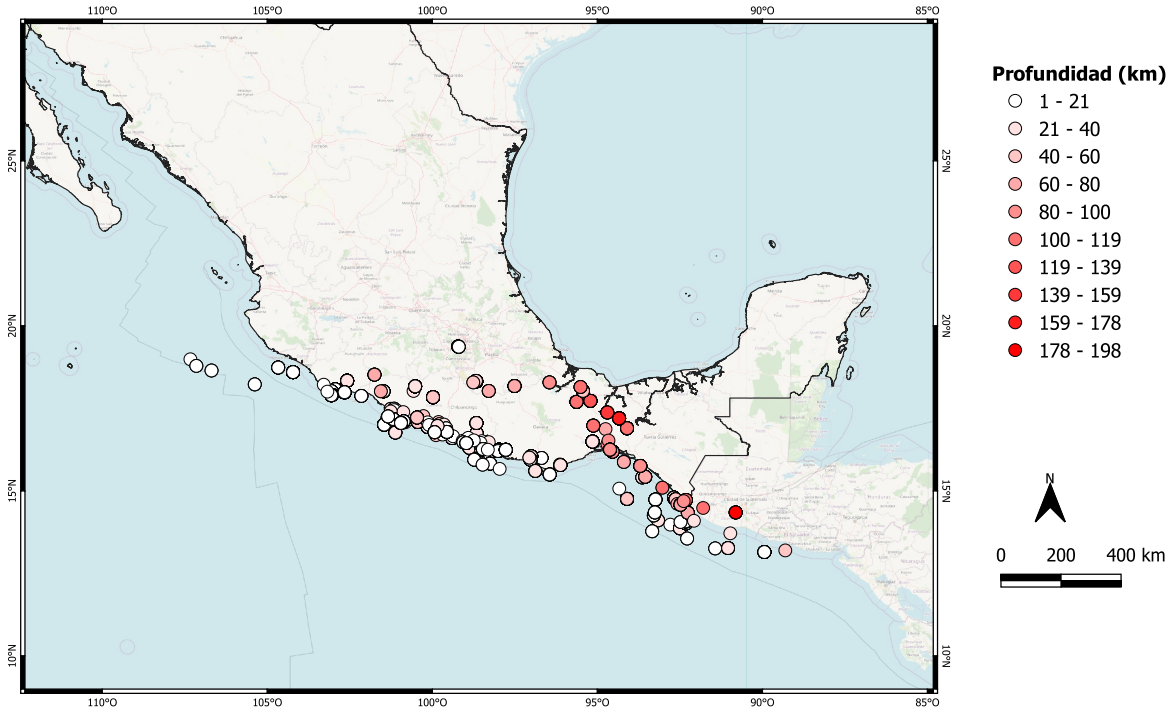

Figura 3.3: Epicentros de los sismos recopilados en la base de datos, clasificados por profundidad hipocentral

observar que la profundidad hipocentral aumenta a medida que se adentra hacia la placa Norteamericana.

Las estaciones sísmicas están distribuidas en la zona centro y sur del país (ver Figura 3.4), donde el registro más cercano a un sismo se encuentra a 1.14 kilómetros, mientras que el más alejado está a 1241.4 kilómetros. El máximo PGA registrado en la base de datos es de 1073.5 gales, mientras que el menor es de 0.16 gales.

Es importante conocer la distribución de valores en la base de datos sísmicos debido a que otorga una primera aproximación al problema y se pueden elegir algoritmos cuyo funcionamiento se adapte de manera óptima a la estructura de los datos. La distribución de datos general de la base de datos se encuentra en la Figura 3.5, en donde se aprecia una alta asimetría con un sesgo hacia los valores pequeños, constituyendo la mayoría de estos, mientras que los valores grandes son la minoría.

La función de probabilidad que describe la base de datos puede aproximarse a la función de densidad de probabilidad Pareto, de donde parte la "regla de Pareto", que explica que el 80 % de los fenómenos observados son explicados por el 20 % de las causas (Reiss & Thomas, 2007). Si bien la base de datos utilizada no cumple con esta regla de

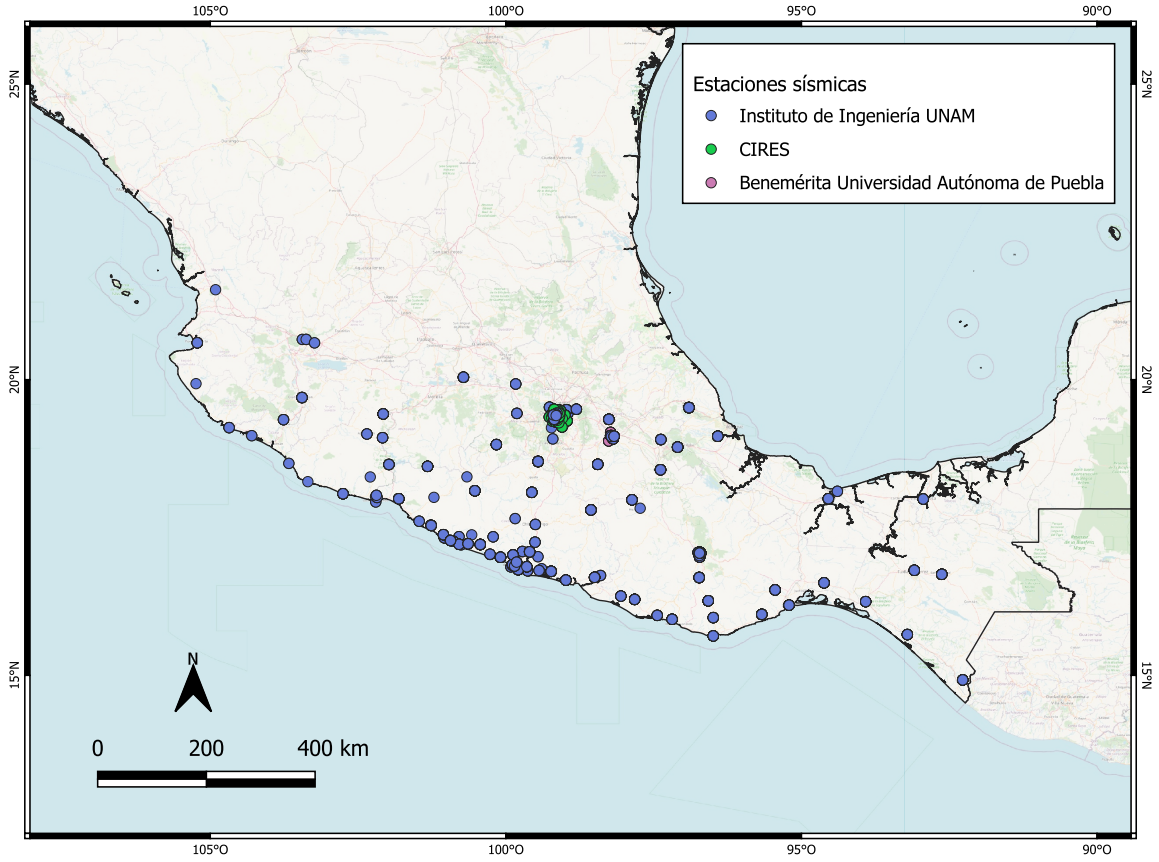

Figura 3.4: Estaciones sísmicas

manera exacta, un análisis estadístico rápido indica que aproximadamente el 16 % de los datos se encuentra por encima del percentil 84. Este valor muestra que la distribución es ligeramente más sesgada que la que explica la regla de Pareto, pero la sigue de manera muy cercana. La función de densidad de probabilidad Pareto está dada por la siguiente fórmula, donde α controla el sesgo:

$$f(x) = \begin{cases} \frac{\alpha x_m^\alpha}{x^{\alpha+1}} & x \geq x_m \\ 0 & x < x_m \end{cases} \tag{3.4}$$

donde x_m es el valor mínimo posible (y estrictamente positivo) de x, la variable que se modela.

Analizando la correlación lineal entre variables, en la Figura 3.6 se muestra la matriz de correlación de Pearson, que indica el nivel de relación lineal que tiene una variable con

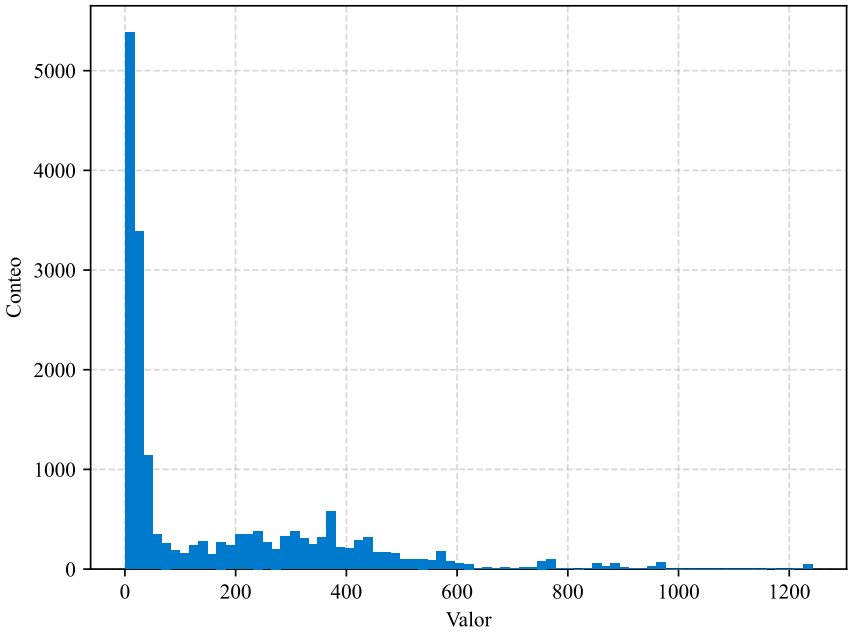

Figura 3.5: Histograma de la base de datos en general. Se aprecia una distribución aproximadamente Pareto, por lo que se requerirán procedimientos de normalización para el correcto funcionamiento de algunos algoritmos.

otra, teniendo rangos de -1 a 1, donde 1 es una relación lineal directa y -1 es una relación lineal inversa. Se puede observar que en general existe una correlación lineal cercana a cero en la mayor parte de las variables (cada una representada por un pixel), lo que indica que no existe correlación y los algoritmos lineales pueden no adaptarse correctamente a los datos al no captar las relaciones reales que existen. Las variables presentes en la matriz de correlación son:

1. **S**: Tipo de suelo
2. **Vs30**: Velocidad de onda S del suelo
3. **M**: Magnitud del sismo
4. **P**: Profundidad hipocentral
5. **D**: Distancia del epicentro a la estación
6. **R**: Distancia del hipocentro a la estación
7. **PGA**: Aceleración máxima registrada en la estación, en cm/s^2

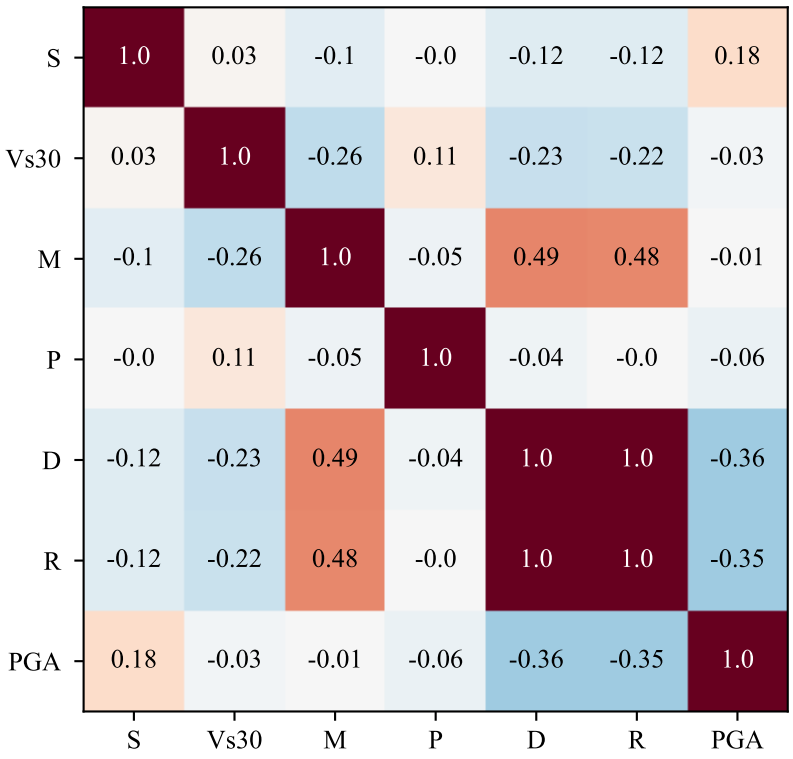

Figura 3.6: Matriz de correlación de Pearson de las variables de la base de datos. Se observa que la mayor parte de las variables tienen una correlación lineal cercana a cero.

Otro tipo de análisis para transformación y visualización de datos con alto número de características es la reducción de dimensionalidad no lineal. Dentro de los algoritmos existentes, el algoritmo *T-stochastic Neighbor Embedding*, t-SNE, es el algoritmo que otorga mejores resultados al mantener una estructura del 98.97 % respecto al conjunto de datos original, por lo que se trata de una representación fiel de la estructura de datos. Se puede observar su alta complejidad que no sigue ningún patrón obvio a primera vista, aunque muestra algunos signos de agrupamiento, por lo que desde este momento se puede inferir que los algoritmos para resolver problemas lineales pueden presentar resultados insatisfactorios y será necesario recurrir a técnicas no paramétricas que logren captar relaciones no lineales.

El algoritmo TSNE está disponible en la biblioteca Scikit-learn de Python, junto con otros algoritmos de reducción de dimensionalidad no lineal y funciones de procesamiento similares, en la clase `sklearn.manifold.TSNE`

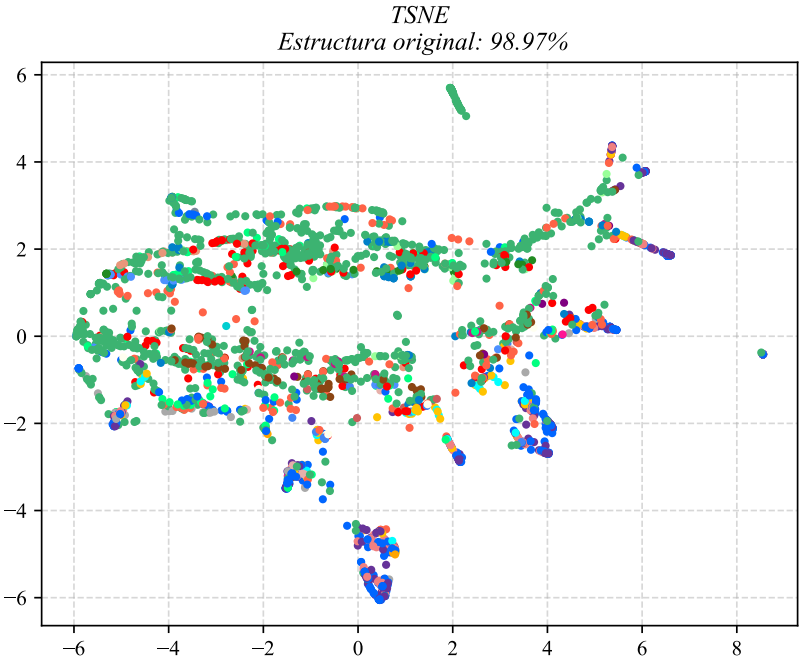

Figura 3.7: Reducción de dimensionalidad de la base de datos a 2 dimensiones con el algoritmo T-stochastic Neighbor Embedding (TSNE) con perplejidad 15 y 250 iteraciones. Se mantiene una estructura del 98.97 %

Los colores en la Figura 3.7 indican la clasificación de los puntos según el tipo de suelo al que pertenecen. La clase con más ocurrencias es del tipo Roca, con 1135 apariciones, por lo que aparece como dominante en color verde en la gráfica.

En la Figura 3.8 se puede observar la relación entre el valor de aceleración máxima

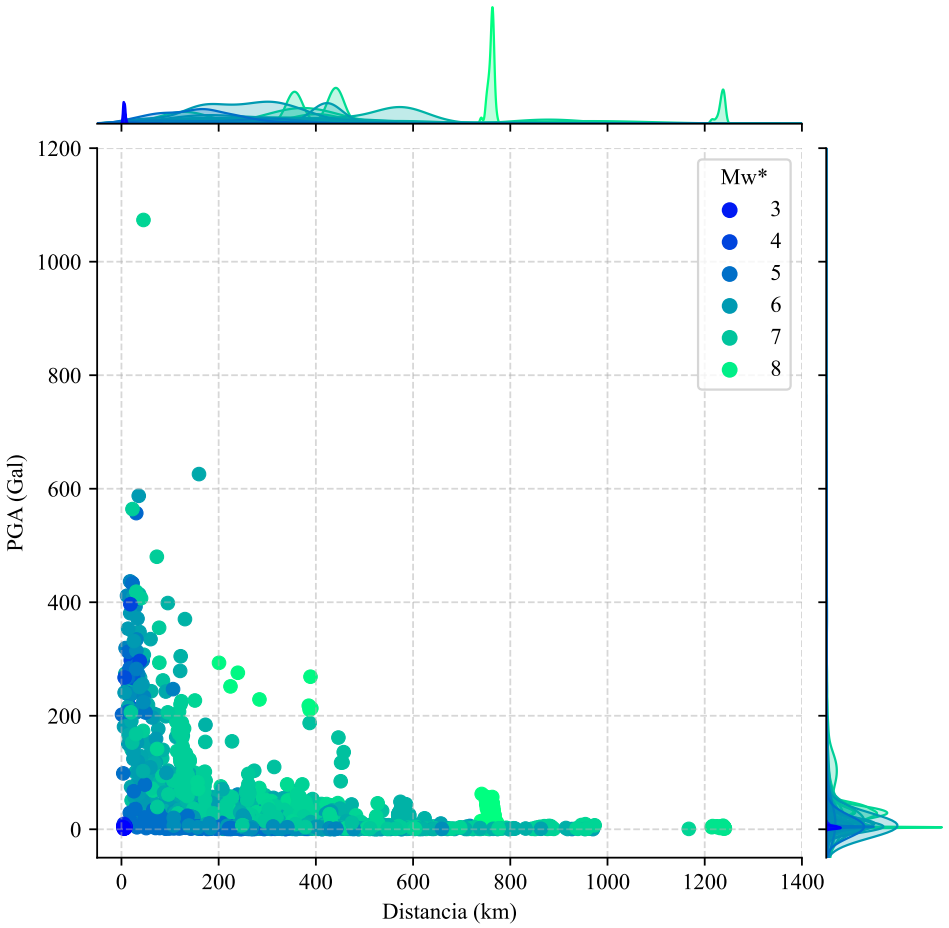

Figura 3.8: Gráfica de distribución de valores de PGA según la distancia de la estación al sismos, clasificados por magnitud sísmica.

medida y la distancia al epicentro del sismo. Se aprecia claramente que el valor de la intensidad registrado decae de manera aproximadamente exponencial, por lo que la mayoría de valores de PGA se encuentran en las cifras pequeñas y disminuye conforme la distancia aumenta. Existe presencia de algunos datos fuera de la tendencia (*outliers*) con un PGA por encima de los 1000 cm/s^2 en distancias muy cortas, pero se observa que se registraron para un sismo de $M_W = 7$, lo cual explica su valor alto. También se observan estaciones cuya distancia al sismo es ligeramente mayor a 1000 km. Finalmente, existe una concentración de valores muy pequeños de PGA relacionados a distancias igualmente pequeñas, las cuales indican los registros de sismos locales en el Valle de México registrados por las estaciones de la zona, los cuales cuentan con magnitudes usualmente menores a $M_W = 3$.

Los *outliers* son pocos en comparación a los datos dentro de la tendencia, por lo que su afectación a la predicción de los modelos es mínima. Además algunos de los algoritmos

que se probarán más adelante son robustos ante *outliers*, por lo que también funcionan como prueba de su adaptabilidad. En el caso de los algoritmos altamente susceptibles a valores muy alejados de la media, se aplican procesos de escalamiento para su correcto funcionamiento, con lo que se espera que su afectación sea mínima, y dado que no se trata de datos erróneos, sino simplemente poco comunes, se decidió mantenerlos en el análisis.

En este capítulo se ha detallado el proceso de recopilación, unificación y análisis exploratorio de los datos utilizados en esta investigación, logrando consolidar una base de datos robusta y representativa, proveniente de diversas fuentes como el Instituto de Ingeniería UNAM y el Centro de Instrumentación y Registro Sísmico. Entre los hallazgos más destacados, se encuentra la identificación de patrones de distribución de las aceleraciones sísmicas y su correlación con variables clave como la distancia al epicentro del sismo, lo cual ha permitido una visualización del comportamiento de la variable objetivo. Estos resultados preliminares sientan las bases para el desarrollo de la metodología que se empleará en la implementación y evaluación de los algoritmos de machine learning. En el próximo capítulo, se describirán detalladamente los procedimientos y técnicas utilizados para preparar los datos, entrenar los modelos y evaluar su desempeño, asegurando así la validez y precisión de las predicciones obtenidas.

Capítulo 4

Metodología

En esta sección se presenta el procedimiento detallado que se llevó a cabo para la comparación de algoritmos de aprendizaje automático. Estos algoritmos reciben como parámetros diversos elementos de los eventos sísmicos y predicen un valor de aceleración máxima. Como ya se mencionó en capítulos anteriores, los acelerogramas utilizados provienen de bases de datos de la Red Acelerográfica del Instituto de Ingeniería UNAM (RAII-UNAM) y el Centro de Instrumentación y Registro Sísmico (CIRES), con sismos ocurridos desde el 5 de enero de 1971 hasta el 14 de diciembre de 2023.

Los datos crudos constan de archivos estándar de aceleración, en un formato llamado ASA2.0, el cual contiene en las primeras 109 líneas un encabezado con información acerca del evento y la estación sísmica a la que corresponde ese acelerograma; debajo del encabezado se encuentra la serie de tiempo registrada por dicha estación en distintos canales, los cuales pueden variar según el sensor, pero generalmente son triaxiales ortogonales. Los registros de aceleración son procesados con funciones personalizadas de Python, con el objetivo de obtener la mayor cantidad de información posible para entrenar el modelo de machine learning. Estos datos incluyen parámetros propios del sismo y características del suelo donde se registró el acelerograma.

Los datos extraídos de los archivos de aceleración son almacenados en una tabla para su preproceso y alimentación al modelo. Se probaron múltiples modelos y algoritmos para evaluar su desempeño ante datos sísmicos, como métodos de descomposición, modelos lineales y árboles de decisión simples y compuestos.

El código de Python fue ejecutado en un entorno remoto de Google Colaboratory utilizando un CPU con las características especificadas en la Tabla 4.1, cuyo diseño otorga

Tabla 4.1: Características del CPU en el entorno remoto

Característica	Descripción
Modelo	Intel(R) Xeon(R) CPU @ 2.20GHz
Número de CPUs	2
Hilos por núcleo	2
Núcleos por socket	1
Tipo de virtualización	Full Virtualization
Tamaños de caché	L1: 32 KiB, L2: 256 KiB, L3: 55 MiB
Flags	AVX, AVX2, SSE4.1, SSE4.2, FMA, etc.
Vulnerabilidades	Diversas; con mitigaciones parciales y vulnerabilidades

un buen desempeño con tareas que requieren un procesamiento numérico intensivo, como lo es el machine learning. Esta computadora además cuenta con 12 GB de memoria RAM.

Después de evaluar los algoritmos y optimizar la selección de hiperparámetros, el modelo con una menor error de predicción se evalúa para verificar su precisión mediante una comparación con datos reales, así como su capacidad de predicción ante eventos hipotéticos. Todos los algoritmos, funciones de transformación y procesamiento previo son implementados desde la biblioteca de Python `Scikit-learn` en su versión `1.4.2`.

4.1. Diseño de funciones para extracción de datos

Los archivos contienen información sobre aceleración, posición del sensor, coordenadas del epicentro y demás características del evento registrado. Esta información debe ser extraída con funciones automáticas para eficientizar el proceso, por lo que se diseñó una función en Python que lee y extrae los datos necesarios del encabezado, sección que contiene la información en una forma estructurada a lo largo de todos los archivos.

Cada archivo corresponde a un acelerograma de un sismo, pudiendo existir cientos de registros para el mismo sismo. Además de la extracción de datos del encabezado, se calcula la aceleración máxima a partir del acelerograma, tomando el valor máximo absoluto de los primeros tres canales disponibles en la serie de tiempo (si existen más de tres canales, usualmente son acelerómetros la misma localización pero en un pozo profundo). La aceleración máxima se calcula como lo indica la siguiente ecuación:

$$PGA = max(A_x, A_y, A_z) \tag{4.1}$$

donde A_x, A_y y A_z son las aceleraciones máximas de las componentes horizontales y la

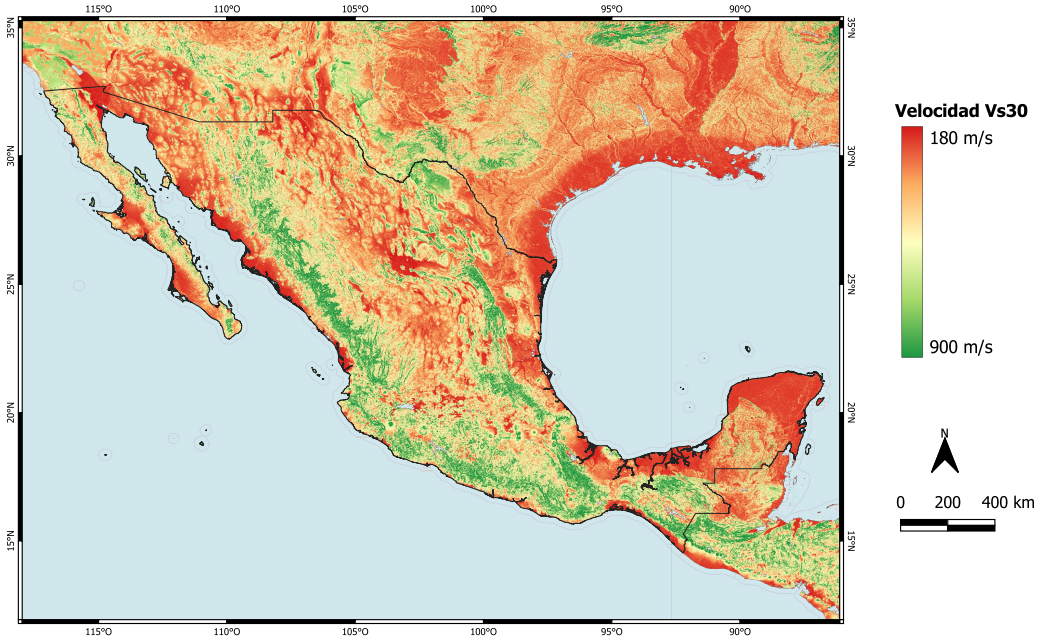

Figura 4.1: Valores de velocidad de onda S (V_{S30}) utilizados para el análisis. Elaboración propia con datos de USGS (2020).

vertical.

Las distancias entre el epicentro del sismo se calcula utilizando geometría plana simple, pues la utilización de geometría esférica representa un erro que puede considerarse despreciable, siendo de apenas 1 km en las distancias más altas, y nulo en las distancias más cortas. De esta forma, la distancia entre puntos es calculada de la siguiente forma:

$$D = \sqrt{|Lat_e - Lat_s|^2 + |Lon_e - Lon_s|^2} \tag{4.2}$$

donde (Lat_e, Lon_e) son las coordenadas del epicentro y (Lat_s, Lon_s) son las coordenadas de la estación sísmica.

De manera similar, la distancia hipocentral se calcula utilizando la profundidad P y la distancia epicentral D previamente calculada, de tal manera que:

$$R = \sqrt{P^2 + D^2} \tag{4.3}$$

En cuanto a los valores de V_{S30}, se mapea el valor correspondiente a cada estación según el ráster de USGS (2020) (ver Figura 4.1) utilizando el software QGis para facilitar

el proceso, exportando los datos finales a un archivo CSV.

4.2. Limpieza de datos

Posterior a la recopilación de datos en una tabla que comprenda la información de todos los registros, es necesario corroborar la calidad de dicha información, por lo que se debe realizar un análisis exploratorio con el objetivo de mantener solamente los datos que contribuirán de manera positiva al modelo y eliminar lo más posible los datos o registros erróneos. La mayoría de los archivos ASA2.0 cuentan con una medida de la calidad del acelerograma, siendo, de excelente a muy mala: A, B, J, X. Para garantizar un entrenamiento preciso, se eligieron únicamente los registros con con calidad B o mayor (la calidad B únicamente se refiere a la falta de hora en la primera muestra, pero la medición de aceleración del sensor es correcta). La descripción detallada de los tipos de calidad disponibles según RAII-UNAM (2023) es el siguiente:

- **A**: Registro digital completo con tiempo absoluto correcto.
- **B**: Registro digital completo, carece de tiempo absoluto.
- **J**: Registro analógico al que le falta una parte al inicio sin tiempo absoluto. Digitalización semiautomática o manual.
- **X**: Registro incompleto en su parte intensa, o con muchos glitches, o película atorada, o dudoso por alguna falla del aparato. En general, un acelerograma no confiable que solo permite tener una idea aproximada de los valores máximos.
- **S/C**: sin clasificación (no hay datos sobre la calidad en el archivo)

Algunas de las tareas de limpieza que fueron necesarias para mejorar la calidad de la base de datos incluyen, pero no está limitado a:

- Revisión de la calidad y selección del acelerograma según las clasificaciones A, B, J, X y S/C
- Homogeneización manual de tipos de suelo debido a errores o discrepancia de lectura y/o puntuación (e.g. 'Blando' y 'Suelo blando')
- Eliminación de datos de acelerogramas con saturación, digitalización imprecisa o poca confiabilidad en general.

Los archivos obtenidos se componen de 2192 con calidad A, 313 con calidad B, 57 con

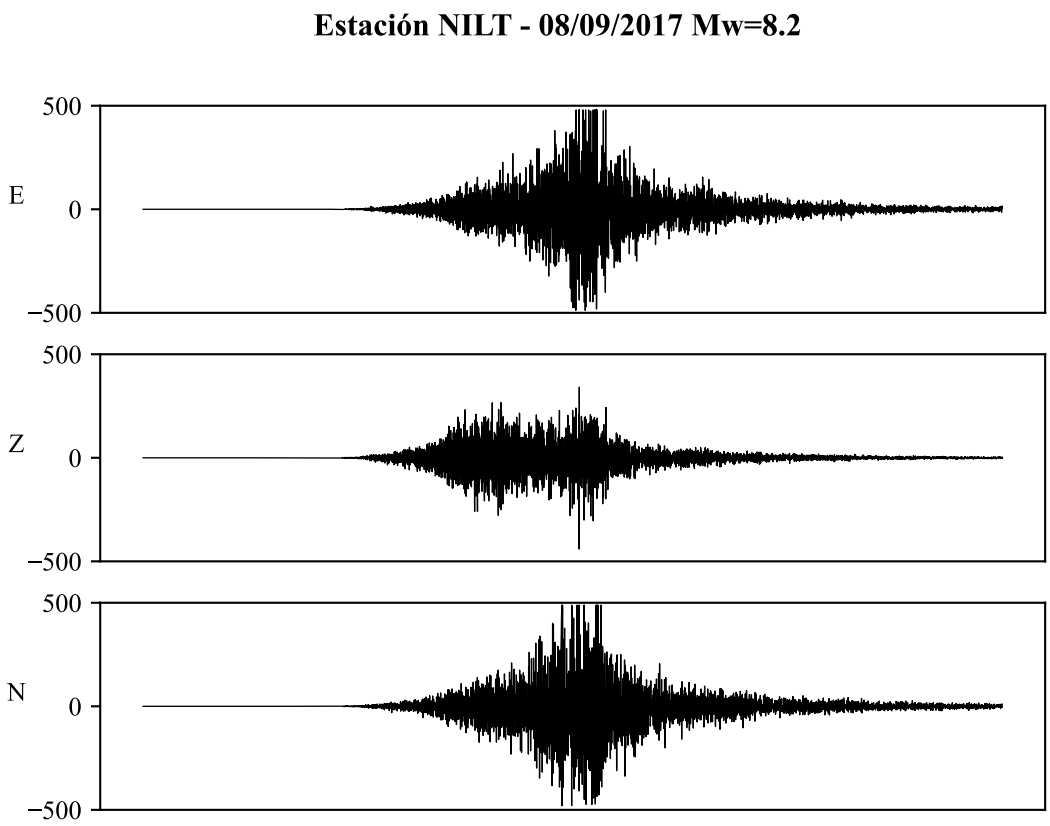

Figura 4.2: Único registro con calidad X de la base de datos. Este registro pertenece al medido por una estación del Instituto de Ingeniería UNAM con clave NILT, localizada en el Colegio de Bachilleres de Niltepec, Oaxaca. La estación NILT se encuentra a 208.59 kilómetros del epicentro del sismo, registrando un PGA máximo de -488.63 gales. Los canales horizontales se encuentran saturados, por lo que la medición no representa el valor de aceleración real en el sitio y no puede usarse para fines de entrenamiento de los modelos de aprendizaje automático.

calidad J y 1 con calidad X. Los archivos restantes no cuentan con información acerca de su calidad, siendo un total de 147 registros sin detalles sobre su calidad. Se eligió descartar únicamente los registros con calidad X y S/C, con lo que la cantidad de datos se reduce a 2562. En la Figura 4.2 se muestra el único registro con calidad X encontrado, el cual se clasifica así debido a la saturación de sus canales horizontales, por lo que el valor registrado no representa el valor real. Finalmente, después de la unificación de los tipos de suelo en los archivos ASA2.0, se cuenta con 27 clases diferentes.

4.3. Preparación de datos para el entrenamiento del modelo

Debido a que la base de datos extraídos a partir de los archivos cuenta con múltiples variables no relevantes para el entrenamiento del modelo o con una correlación muy alta, se deben seleccionar solamente aquellos datos que contribuyan a un resultado significativo en la predicción. Variables como la clave de la estación, la fecha y hora del sismo y el orden de los canales de medición en el registro, se vuelven irrelevantes en el proceso de predicción, por lo que son removidas para reducir el tamaño del conjunto de datos y optimizar la eficiencia computacional. Así, las columnas cuyos datos fueron conservados son los siguientes:

- Tipo de suelo en el que se encuentra el sensor
- Velocidad de onda S en m/s
- Magnitud del sismo
- Profundidad hipocentral en kilómetros
- Distancia del epicentro al sensor en kilómetros
- Valor máximo de PGA en los 3 primeros canales en gales (cm/s^2)

Debido a que el tipo de suelo se trata de una variable textual (por ejemplo: granito, transición, limo arenoso, depósitos lacustres, arcilla, etc.), no puede alimentarse a algunos de los modelos que solamente funcionan con variables numéricas. Para solucionar este tipo de problemas, la biblioteca Scikit-learn cuenta con una serie de funciones especializadas en la transformación de variables en su clase `sklearn.preprocessing`. Algunas de estas funciones son `OrdinalEncoder`, `OneHotEncoder` y `TargetEncoder`, cada una con diferente funcionamiento según el objetivo perseguido.

En el caso de los datos de tipo de suelo, se decidió utilizar la función `TargetEncoder` debido a que soluciona el problema de `OneHotEncoder` al crear demasiadas columnas binarias extra, o el de `OrdinalEncoder` de asignar valores numéricos arbitrarios a las variables cuya importancia será tratada según su valor numérico, lo que ocasiona un sesgo considerable hacia variables con valores mayores por simple ordinalidad.

`TargetEncoder`, por otro lado, utiliza información de la variable objetivo para asignarle un valor numérico según el valor promedio del objetivo asociado a esa clase. Para evitar el filtrado de información durante la codificación de las variables categóricas con

Tabla 4.2: Codificaciones numéricas obtenidas por `TargetEncoder`

Tipo de suelo	**Codificación numérica**
Alto riesgo sísmico	33.69083
Aluvial	44.27667
Arcilla	32.82302
Arenoso, limoso, compacto	26.32703
Blando	24.81029
Deposito barra	36.36694
Depósitos lacustres	33.22859
Edificio	32.63319
Estructura	29.08659
Granito	35.61953
Granito alterado	32.52631
Limo arenoso	54.49238
Mediana compresibilidad	32.40859
Montado sobre tubería de concreto	31.32469
Monumento arqueológico	31.95934
No clasificado	38.84710
Presa concreto	32.56355
Relleno semicompactado	31.28962
Roca	33.00142
Suelo	25.58752
Suelo duro	27.12336
Terreno blando, material compresible	32.90196
Terreno estratificado	31.57493
Terreno firme, materiales compactos	29.23775
Terreno libre	29.75313
Transición	28.63127
Travertino	28.52156

`TargetEncoder`, scikit-learn utiliza un proceso con validación cruzada que separa los datos de entrenamiento en k partes, de tal manera que cada parte sea entrenada con las categorías vistas por las $k-1$ partes restantes (Pedregosa et al., 2011) y se evite lo más posible la filtración de datos de la variable objetivo.

De esta manera, cada tipo de suelo en la base de datos es asignada a un valor numérico asociado al valor de la variable objetivo, dando una importancia proporcional al valor de PGA correspondiente. Los valores asignados a cada variable se encuentran en la Tabla 4.2.

4.4. Implementación de los algoritmos

La propuesta de algoritmos para entrenarse con los datos incluye algunos de los más comunes en su tipo, desde modelos lineales hasta árboles de decisión altamente complejos. Esto asegura que se cubren distintos acercamientos para encontrar un buen desempeño, lo que se traduce en un decisión más objetiva al momento de elegir el mejor. De manera general, los algoritmos explorados fueron los siguientes:

- Modelos lineales
  - Regresión lineal ordinaria por mínimos cuadrados
  - Regresión lineal de cresta con regularización L2 o de Tikhonov
  - Regresión con redes elásticas con regularización L1 y L2
- Árboles de decisión
  - Árboles de decisión simples
  - Bosques aleatorios (*Random Forest*)
  - Impulso por gradiente (*Gradient Boosting*)
  - Impulso por gradiente extremo (*Extreme Gradient Boosting*)
- Reducción de dimensionalidad
  - Análisis de componentes principales (PCA, *Principal Component Analysis*)
  - Mínimos cuadrados parciales (PLS, *Partial Least Squares*)

Para asegurar que los algoritmos creen modelos con predicciones precisas y que reflejen de manera real su capacidad de adaptarse a los datos, es necesario que se dividan en un conjunto de entrenamiento (80 %) del total y un conjunto de prueba (20 % del total), de manera que el algoritmo se evalúe en datos aún no vistos. Además, para optimizar los hiperparámetros de cada método se utilizó la función `RandomizedSearchCV` del módulo `sklearn.model_selection`, el cual prueba distintas combinaciones aleatorias de hiperparámetros a partir de una matriz de entrada, evaluando el desempeño del modelo con cada combinación. A diferencia de otros métodos de búsqueda como `GridSearchCV`, que prueba todas las combinaciones posibles, el número de pruebas `RandomizedSearchCV` está controlado por un número de iteraciones fijo, por lo que su costo computacional puede ser controlado en mayor medida. Otra ventaja de `RandomizedSearchCV` es que puede encon-

Tabla 4.3: Hiperparámetros para la optimización de algoritmos en RandomizedSearchCV

Hiperparámetro	**Valor**
estimator	En función del estimador actual
param_distributions	En función del estimador actual
n_iter	10
scoring	r2
error_score	raise
return_train_score	True
verbose	0
cv	KFold(n_splits=10, shuffle=True)
random_state	42
refit	False
n_jobs	-1

trar soluciones óptimas en espacios que serían inalcanzables por `GridSearchCV` (Pedregosa et al., 2011).

Todos los algoritmos son evaluados a través de la búsqueda de hiperparámetros mediante `RandomizedSearchCV` con los valores de la Tabla 4.3.

Estos parámetros aseguran que cada conjunto aleatorio de hiperparámetros sea evaluado 10 veces en cada iteración, dando un total de 100 evaluaciones con validación cruzada por cada combinación de hiperparámetros utilizada. El proceso de validación cruzada consiste en dividir los datos en 10 partes. En cada una de las 10 evaluaciones, el modelo se entrena en 9 de estas partes y se evalúa en la parte restante. Este ciclo se repite cambiando la parte de evaluación en cada iteración, de modo que cada parte se utiliza una vez como conjunto de evaluación. Este método permite que el algoritmo se entrene múltiples veces en distintos subconjuntos de los datos, mejorando su capacidad de generalización y su robustez frente a datos no vistos. Esto se logra gracias al uso de la función `KFold(n_splits=10)`.

Se abordará con mayor detalle la implementación de cada algoritmo en las secciones posteriores, pues cada uno cuenta con sus propias condiciones y ajustes particulares que le permiten llegar a la solución óptima a través de la validación cruzada.

4.4.1. Modelos lineales

Regresión lineal ordinaria

Este algoritmo se trata de una regresión lineal por mínimos cuadrados, mediante la cual se estima la variables objetivo a partir de las relaciones lineales entre variables. Este

estimador también puede llevar a cabo una regresión lineal multivariable (el caso de este proyecto).

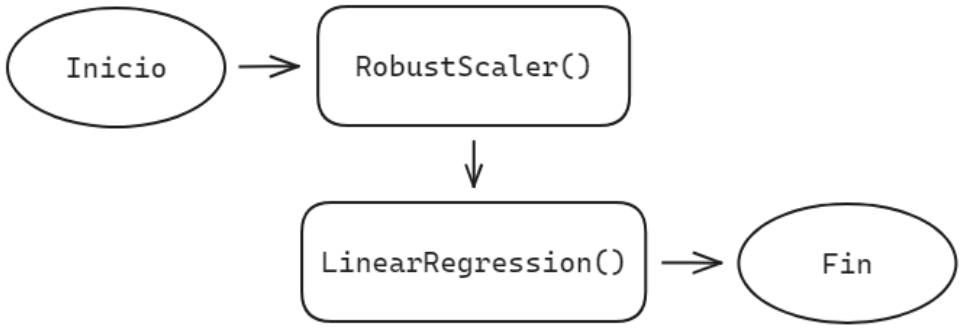

Figura 4.3: Diagrama de flujo de la implementación de la regresión lineal ordinaria en Scikit-learn

La clase encargada de este algoritmo en Scikit-learn se encuentra en la clase `sklearn.linear_model.LinearRegression`. Dado que se planea estimar las relaciones entre variables, este algoritmo requiere de un escalamiento de variables para que posean valores comparables, por lo que se hace uso de un *pipeline*, el cual es un método de Scikit-learn de enlazar varios procesos en un mismo objeto, como se ejemplifica en el diagrama de la Figura 4.3.

El algoritmo de regresión lineal no requiere de ningún hiperparámetro de optimización, por lo que probablemente es el modelo más sencillo de implementar en Scikit-learn.

Regresión de cresta

La regresión de cresta es una alternativa a la regresión lineal simple para solucionar su susceptibilidad a la colinealidad de los datos. Dado que una colinealidad puede afectar el desempeño de la regresión lineal, la regresión de cresta implementa un factor de regularización α para controlar el tamaño de los coeficientes en el ajuste. Mientras mayor sea el factor de regularización, mayor será la robustez ante la colinealidad.

Tabla 4.4: Matriz de hiperparámetros para la regresión de cresta, en formato (mínimo : máximo : número de elementos).

Hiperparámetro	**Valor**
alpha	(0.1 : 5 : 5)
tol	(log(-1 : 1.5) : 5)

Este objeto toma como hiperparámetros de optimización el valor de regularización y la tolerancia, que especifica el nivel de precisión requerido en la solución. Así, la matriz de

Tabla 4.5: Matriz de hiperparámetros para la regresión con red elástica

Hiperparámetro	**Valor**
alpha	(0.1 : 5 : 5)
l1_ratio	(0.1 : 1 : 5)
max_iter	(500 : 10000 : 5)

hiperparámetros de optimización se presenta en la Tabla 4.4, donde el vector de búsqueda se representa como (mínimo : máximo : número de elementos).

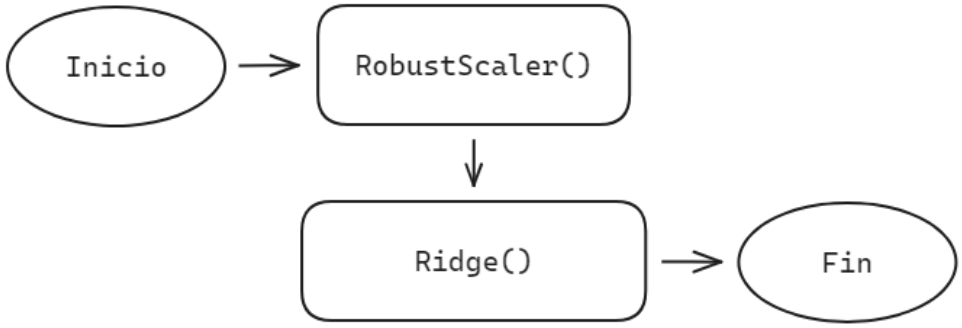

Figura 4.4: Diagrama de flujo de la implementación de la regresión lineal de cresta en Scikit-learn

En Scikit-learn, el algoritmo de regresión de cresta, o regresión lineal con regularización de Tikhonov, se implementa a través de la clase `sklearn.linear_model.Ridge`, el cual también requiere del escalamiento de sus datos de manera previa, de modo que se implementa a través de un *pipeline*, descrito en la Figura 4.4.

Regresión con red elástica

Este tipo de regresión lineal combina las penalizaciones encontradas en los estimadores de cresta y del modelo Lasso. Esto garantiza que se evalúa el modelo bajo las regularizaciones L1 y L2, lo cual controla el tamaño de los coeficientes, de forma que los elimina si son poco relevantes o los regulariza si son demasiado grandes. Los hiperparámetros que se buscan optimizar en este modelo es la tasa de regularización L1/L2, el máximo número de iteraciones del modelo y una constante de multiplicación los términos de regularización. La matriz de optimización de hiperparámetros se presenta en la Tabla 4.5:

Al igual que los modelos lineales anteriores, este algoritmo requiere de un escalamiento de datos antes de su entrenamiento, por lo que su implementación es bastante similar a los dos casos anteriores, siguiendo la estructura de la Figura 4.5.

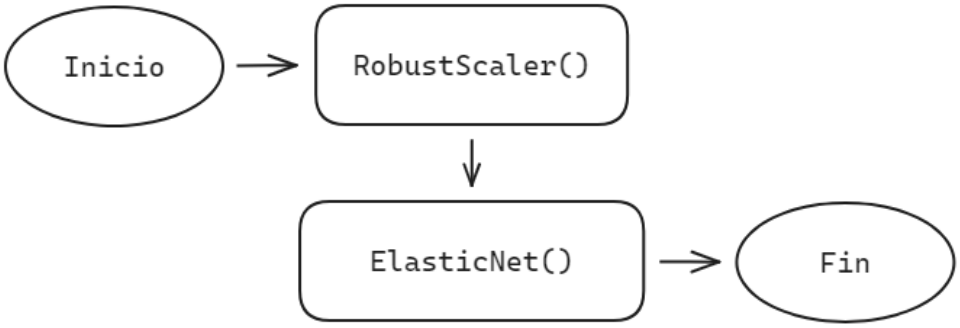

Figura 4.5: Diagrama de flujo de la implementación de la regresión de red elástica en Scikit-learn

Tabla 4.6: Matriz de hiperparámetros para árboles de decisión simples

Hiperparámetro	**Valor**
min_samples_split	(2 : 8 : 5)
min_samples_leaf	(1 : 9 : 5)

4.4.2. Árboles de decisión simples

Los árboles de decisión son algoritmos que utilizan reglas booleanas sencillas para llegar a un resultado, las cuales se infieren a partir de los datos de entrada, por lo que su proceso es fácilmente interpretable. Aunque los árboles de decisión tienden a tener buena precisión, una de sus grandes desventajas es que pueden caer en la sobreestimación de los datos si el parámetro de profundidad no se elige correctamente (Pedregosa et al., 2011).

Para implementar el algoritmo de regresión por árboles de decisión en Scikit-learn, se utiliza la clase `sklearn.tree.DecisionTreeRegressor`. El árbol de decisión toma como hiperparámetros el criterio de medición de error, siendo error cuadrático medio en este caso, el número de elementos mínimos para pasar a otra rama y el mínimo para considerarse una hoja. Cuanto menor sea la cantidad de elementos por hoja, más preciso será, pero también más propenso a sobreestimación. La matriz de hiperparámetros para el árbol de decisión se encuentra en la Tabla 4.6.

Otra ventaja de los árboles de decisión es que dado que no aproximan ninguna función matemática analítica y su estimación se basa en decisiones simples, sí son algoritmos invariantes a la escala , por lo que su uso no requiere de ningún *pipeline* como los algoritmos anteriores, de manera que su implementación es muy simple.

Este modelo ofrece resultados mucho mejores que los algoritmos revisados anteriormente con un costo computacional muy bajo, como se verá en los resultados más adelante.

4.4.3. Métodos de conjunto

Los métodos de conjunto incluyen aquellos que combinan las predicciones de múltiples estimadores simples para mejorar la generalización del modelo, de tal manera que su robustez se ve incrementada (Pedregosa et al., 2011). Se prueban dos tipos de métodos de conjunto: bosques aleatorios y árboles impulsados por gradiente, tanto en su versión normal como en la versión con uso de histogramas. Dada la naturaleza de estos estimadores y los resultados observados por los árboles de decisión simples, se espera que tengan el mejor desempeño.

Bosques aleatorios

La implementación de los bosques aleatorios en Scikit-learn se basa en el algoritmo de Breiman (2001), donde cada árbol es construido a través de una muestra tomada aleatoriamente del set de datos de entrenamiento. La puntuación final del bosque aleatorio es el promedio de la puntuación de todos los árboles de decisión evaluados en muestras aleatorias distintas (Pedregosa et al., 2011).

Tabla 4.7: Matriz de hiperparámetros para bosques aleatorios

Hiperparámetro	**Valor**
max_depth	(2 : 6 : 20)
min_samples_leaf	(1 : 10 : 10)
min_samples_split	(2 : 6 : 4)
max_leaf_nodes	(50 : 100 : 10)
n_estimators	(20 : 100 : 10)

El uso de bosques aleatorios en Scikit-learn es muy directo, pues no requiere de ningún preprocesamiento, escalamiento ni codificación de variables categóricas (pero en este caso sí se realizó una codificación con fines de mantener los mismos datos de entrenamiento para todos los algoritmos). Además, los hiperparámetros que requieren optimización en los bosques aleatorios son pocos y fáciles de comprender, siendo éstos muy similares a los árboles de decisión simples, como se observa en la Tabla 4.7.

Dado que el algoritmo no requiere procesamiento previo, su utilización e implementación simplemente requiere declarar una instancia con los hiperparámetros fijos deseados, siendo únicamente el criterio de puntuación de los árboles y el número de hilos del procesador que se utilizarán durante el entrenamiento.

Árboles impulsados por gradiente

La implementación de este algoritmo en Scikit-learn se encuentra, al igual de los bosques aleatorios, en la clase `ensemble`, donde se implementan dos tipos de este algoritmo, uno 'normal' y otro basado en la agrupación por histograma. Este último, según la propia documentación y como se mencionó en la Sección 2.5.3, puede ser órdenes de magnitud más rápido cuando se trata con más de diez mil datos, por lo que se eligió este para ser entrenado con los datos sísmicos.

Tabla 4.8: Matriz de hiperparámetros para impulso por gradiente

Hiperparámetro	**Valor**
learning_rate	(0.05 : 0.3 : 20)
max_iter	(50 : 150 : 10)
max_leaf_nodes	(2 : 20 : 10)
min_samples_leaf	(1 : 6 : 10)
l2_regularization	(log(-2 : 1) : 5)

Este algoritmo se puede encontrar como `HistGradientBoostingRegressor`, cuyos principales hiperparámetros por optimizar son muy similares a los de un árbol de decisión (por tratarse de un conjunto de árboles de decisión inherentemente), lo cuales se describen en la Tabla 4.8.

Al igual que `RandomForestRegressor`, su implementación simplemente requiere llamar a una instancia del estimador `HistGradientBoostingRegressor` y entrenarlo con los datos deseados.

Impulso por gradiente extremo

El algoritmo de impulso de gradiente extremo no forma parte de Scikit-learn como tal, sino que se encuentra como software independiente en una librería llamada `XGBoost`, pero está diseñada para funcionar de manera similar a Scikit-learn, por lo que su utilización conlleva una curva de aprendizaje muy suave. Además, según Chen y Guestrin (2016), esta implementación del algoritmo está optimizada para ejecutarse de forma paralela si el procesador utilizado tiene las capacidades y de resolver problemas con más de mil millones de muestras.

Este algoritmo toma hiperparámetros propios de la generación de cada árbol y sobre el nivel de aprendizaje que se requiera en cada uno o el nivel de regularización en los coeficientes. La matriz de hiperparámetros de optimización debe ser cuidadosamente elegida, pues de esto dependerá el alcance del algoritmo. De esta manera, la matriz de

Tabla 4.9: Matriz de hiperparámetros para impulso por gradiente extremo

Hiperparámetro	Valor
learning_rate	(0.05 : 0.3 : 20)
max_depth	(3 : 6 : 5)
n_estimators	(20 : 300 : 10)
subsample	(0.5 : 0.8 : 3)
alpha	(0, 0.01, 0.1, 1, 10)
lambda	(0, 0.01, 0.1, 1, 10)

hiperparámetros elegidos se describen en la Tabla 4.9.

El algoritmo de XGBRegressor, aunque no pertenece a Scikit-learn, está programado para apegarse a sus reglas de diseño, por lo que su implementación y uso es exactamente igual que las demás clases pertenecientes a Scikit-learn. La implementación del algoritmo se realiza de la siguiente forma:

4.4.4. Reducción de dimensionalidad

Estos algoritmos se proponen con el fin de reducir el tamaño del conjunto de datos inicial y hacer más eficientes los cálculos de regresión y ajuste. Dado que el conjunto de datos de entrenamiento consta de 7 columnas por 2709 filas, se cuenta con una alta dimensionalidad, lo que puede traer problemas de eficiencia y desempeño en algunos algoritmos, además de poder encontrarse en una condición de redundancia o colinealidad entre sus variables. Este tipo de algoritmos son sensibles a la magnitud de los datos, por lo que se deben modificar para que tengan una escala común y no haya sesgo debido a las distintas escalas.

Análisis de Componentes Principales, PCA

El algoritmo `KernelPCA` está implementado en Scikit-learn en la clase de métodos de descomposición `sklearn.decomposition.KernelPCA`. Esta es una variación con kernel polinomial de PCA, el cual toma como hiperparámetros el número de dimensiones a la cuales se desea reducir el set de datos, la tolerancia en la solución y el grado del polinomio utilizado en su kernel de ajuste.

Dado que este método no es utilizado para visualización en este caso, el número de dimensiones no tiene que ser dos forzosamente, es por esto que la matriz de hiperparámetros para PCA consta de un vector de posibles dimensiones de 5 elementos (de 2 a 6) y 5 niveles de tolerancia, de 1×10^{-5} hasta 1. En la Tabla 4.10 se detallan los hiperparámetros

Tabla 4.10: Matriz de hiperparámetros para PCA

Hiperparámetro	**Valor**
n_components	(2 : 6 : 5)
tol	(log(-5 : 0) : 5)
degree	(2,3,4,5)

utilizados en PCA.

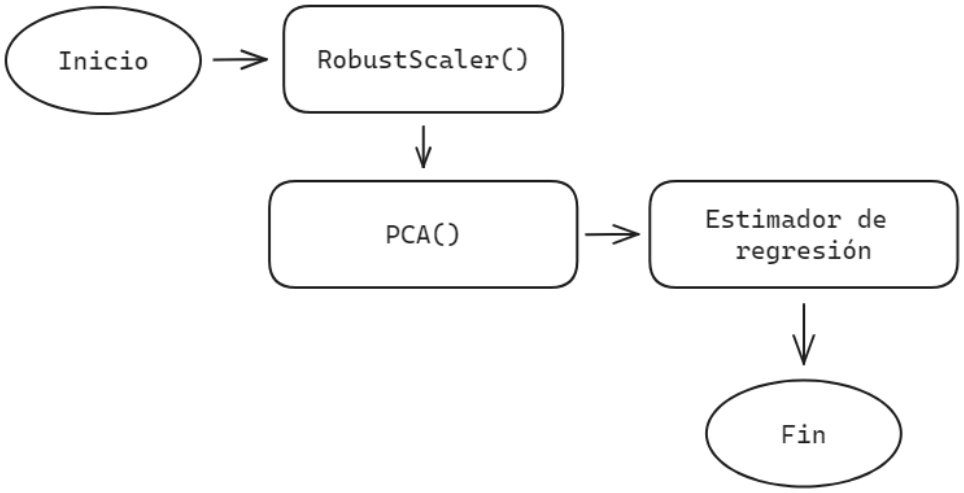

Figura 4.6: Diagrama de flujo de la implementación de PCA en Scikit-learn

Dado que PCA solamente es un algoritmo de reducción de dimensionalidad, no incluye ningún método para realizar regresión por sí mismo, por lo que se utilizó el algoritmo de mejor desempeño para evaluar la viabilidad de una reducción de dimensionalidad y una posible mejora en el desempeño. La utilización de PCA junto con algoritmos de regresión a través de un *pipeline* asegura que los datos estén transformados antes de aplicar una regresión. El diagrama de flujo para la implementación de PCA se ilustra en la Figura 4.6.

Mínimos Cuadrados Parciales, PLS

Este algoritmo se elige bajo la premisa de que el conjunto de datos sísmicos tiene una gran cantidad de variables dependientes y PLS podría ofrecer resultados satisfactorios en comparación con PCA simple.

La implementación de PLS en Scikit-learn se encuentra en la clase de descomposición cruzada `sklearn.cross_decomposition.PLSRegression`, la cual toma como hiperparámetros el número de dimensiones para la descomposición, el máximo de iteraciones durante la regresión y el nivel de tolerancia, como se detalla en la Tabla 4.11.

Tabla 4.11: Matriz de hiperparámetros para PLS

Hiperparámetro	Valor
n_components	(1 : 10 : 10)
tol	(log(-10 : -3) : 10)
max_iter	(5 : 50 : 10)

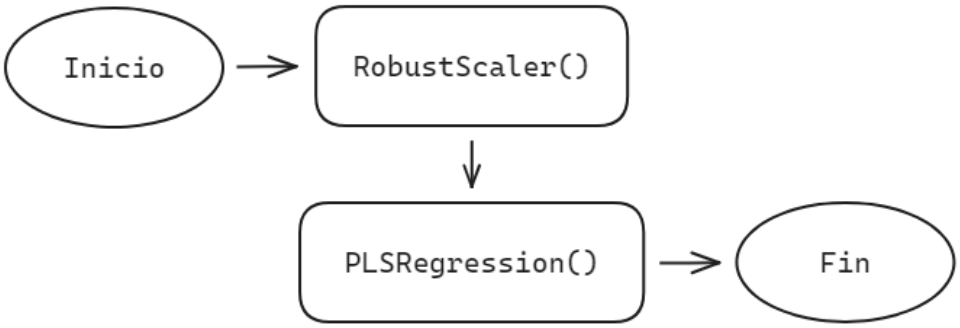

Figura 4.7: Diagrama de flujo de la implementación de PLS en Scikit-learn

Debido a que PLS ya incluye la regresión y predicción dentro de su algoritmo, solamente basta con escalar las variables previamente para asegurar una predicción acertada. De esta manera, el diagrama de implementación para PLS en Scikit-learn se ejemplifica en la Figura 4.7.

4.5. Métricas de desempeño

Con el objetivo de evaluar los modelos bajo una métrica que permita ajustarse mejor al problema propuesto, varios tipos de funciones de pérdida son evaluados. Las métricas evaluadas son el error absoluto medio, error cuadrático medio, error cuadrado medio y el coeficiente de determinación. Cada una de estas métricas ofrece resultados y penalizaciones distintas.

1. **Error absoluto medio**: es una métrica que mide la media de la diferencia absoluta entre los valores reales y los valores calculados, asignando la misma penalización a todos los valores sin importar la magnitud de éstos, es decir, simplemente ofrece una medida de lo alejado que está el modelo de los datos originales (Willmott & Matsuura, 2005). Su función de cálculo es la siguiente:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - f_i| \tag{4.4}$$

2. **Error cuadrado medio**: esta métrica toma la diferencia al cuadrado de los valores predichos y su correspondiente valor real, de manera que la penalización es mayor cuanto más grande sea la diferencia entre valores. Su función de cálculo es la siguiente:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - f_i)^2 \tag{4.5}$$

3. **Error cuadrático medio**: de manera similar al error cuadrado medio, esta métrica toma el cuadrado de la diferencia entre los valores, ofreciendo una penalización mayor a los errores más grandes. La diferencia de este método es que al calcular la raíz cuadrada del error cuadrado se obtiene una métrica en las mismas unidades que la variable objetivo, por lo que es más fácilmente interpretable. El cálculo de esta métrica se realiza según la siguiente ecuación:

$$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - f_i)^2} \tag{4.6}$$

4. **Coeficiente de determinación**: también llamado puntuación R^2, el coeficiente de determinación mide la cantidad de varianza de la variable objetivo que ha sido explicada por las variables independientes del modelo. En otras palabras, indica la calidad del ajuste, lo que se traduce como qué tan probable es que el modelo prediga la muestras aún no vistas de manera correcta (Pedregosa et al., 2011), y a diferencia de las métricas anteriores, mientras mayor sea el coeficiente de determinación, mejor será el modelo ajustado. Se calcula a partir del cociente entre la suma total de error residual y la diferencia total respecto a la media $\bar{y}$ de los datos, con un rango que va de 0 a 1, como se muestra a continuación:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - f_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4.7}$$

Aunque el RMSE puede prestarse a sesgos de interpretación debido a su fórmula de cálculo (Willmott & Matsuura, 2005), se observó que los algoritmos se ordenan de igual forma independientemente de la métrica analizada. Por lo tanto, la selección del mejor algoritmo puede realizarse a través de cualquiera de los errores calculados. Sin embargo, se eligió el RMSE como la principal métrica porque permite observar cuánto mayor es el error respecto al error absoluto medio debido a errores más grandes. Además, en el contexto de

valores de aceleración en el suelo debido a sismos en zonas pobladas, un mayor error de predicción debería penalizarse en mayor medida.

Como métrica adicional, se introduce un factor F que toma en cuenta el coeficiente de determinación y el tiempo de entrenamiento del algoritmo, reduciendo este factor cuanto mayor sea el tiempo de entrenamiento y aumentándolo si la puntuación R^2 es mayor. De esta forma, se obtiene el candidato que ofrece la mejor relación entre precisión y eficiencia computacional (que no necesariamente se trata del mejor candidato absoluto), quedando el cálculo de la siguiente manera:

$$F = \frac{R^2}{t} \tag{4.8}$$

Durante este capítulo se ha descrito de manera detallada los procedimientos y técnicas empleadas para preparar los datos, implementar los algoritmos de machine learning y evaluar su desempeño en la predicción de aceleraciones sísmicas. Se han incluido los pasos de diseño de funciones para la extracción y limpieza de datos, la preparación de los mismos para el entrenamiento de modelos, y la implementación de diversos algoritmos, incluyendo modelos lineales, árboles de decisión y métodos de conjunto. Además, se han definido las métricas de desempeño utilizadas para comparar la eficacia de los modelos, tanto de forma absoluta como relativa.Posteriormente, en el siguiente capítulo se presentarán y analizarán los hallazgos obtenidos a partir de la aplicación de los métodos descritos, permitiendo evaluar su eficacia, ventajas y desventajas, así como un ejemplo de aplicación práctico.

Capítulo 5

Resultados

En este capítulo se presentan los resultados de los modelos evaluados para la predicción de la aceleración máxima del terreno (PGA) debido a un sismo, utilizando datos de aceleración recopilados por diversas instituciones en México. Los resultados incluyen métricas de desempeño, como el error cuadrático medio (RMSE) y el coeficiente de determinación (R^2), además del tiempo necesario para que el algoritmo encuentre y optimice los hiperparámetros, así como el tiempo requerido para su entrenamiento. Adicionalmente, se incorporan gráficas y figuras que ilustran visualmente los resultados de cada modelo, incluyendo gráficos comparativos y tablas con detalles. Como ejemplo de aplicación del mejor modelo identificado, se emplean datos reales de un sismo con magnitud $M_W = 7.2$ para generar mapas comparativos de predicción de aceleración. Estos mapas permiten evaluar el desempeño predictivo del algoritmo y establecer relaciones entre variables que explican adecuadamente el fenómeno estudiado. Finalmente, se destaca la capacidad del modelo para simular la aceleración sísmica máxima en escenarios hipotéticos, generando datos para los eventos propuestos.

Los resultados se encuentran clasificados por el tipo de algoritmo, que son métodos de regresión lineal, descomposición, árboles de decisión y métodos compuestos (árboles de decisión impulsados por gradiente y bosques aleatorios).

Es importante destacar que el tiempo de optimización depende en gran medida de la distribución de hiperparámetros proporcionada y de los recursos computacionales disponibles. Debido a esta variabilidad, solamente se considera el costo computacional para evaluar el rendimiento intrínseco de cada modelo.

A continuación, se presenta un análisis detallado de los resultados para cada uno de

los modelos probados.

5.1. Modelos lineales

Dado que los métodos lineales asumen una relación directa entre las variables explicativas y el resultado sin incorporar procesos complejos, ofrecen resultados rápidos con un bajo costo computacional. No obstante, cuando se aplican a datos que presentan una relación no lineal, es probable que su desempeño se vea considerablemente afectado. Estos métodos se utilizan principalmente como un punto de referencia para comparar el rendimiento de algoritmos más complejos.

5.1.1. Regresión lineal ordinaria

El algoritmo utilizado para la regresión lineal ordinaria es el de *LinearRegression* de Scikit-learn, disponible en la clase `sklearn.linear_model`, que tiene como ventaja su facilidad de implementación, pero está diseñado para un tipo de datos muy específico, por lo que se espera que su resultados en datos sísmicos no sea óptimo.

Como se observa en la gráfica de error de predicción de la Figura 5.1, al graficar la dispersión entre los valores reales y los valores predichos por el algoritmo, se encuentra que un modelo lineal ordinario es incapaz de captar la relación entre variables de los datos originales, por lo que su nivel de error es muy alto, teniendo un RMSE de 55.11 y una puntuación R^2 de solamente 0.255. Un modelo ideal debería contar con una gráfica de error de predicción cuyos datos se encuentren concentrados a lo largo de la línea punteada.

5.1.2. Regresión de cresta

Al ser un tipo de regresión lineal con regularización, la regresión de cresta representa una mejor alternativa a la regresión lineal ordinaria, la cual lo hace más robusto ante la colinealidad entre variables (McDonald, 2009). Debido a que los datos sísmicos utilizados muestran una colinealidad entre algunas de sus variables (ver Figura 3.6), se espera que otorgue mejores resultados que su contraparte ordinaria.

De manera muy similar al caso de la regresión lineal ordinaria, este estimador presenta un nivel de ajuste inutilizable para fines de predicción. Obsérvese la Figura 5.2, donde se presenta la gráfica de error de predicción de la regresión lineal con regularización, la cual muestra que su nivel de ajuste a los datos originales es demasiado pobre, con un valor de RMSE de 55.3 y una puntuación R^2 de 0.2499, mostrando que, aunque es un método

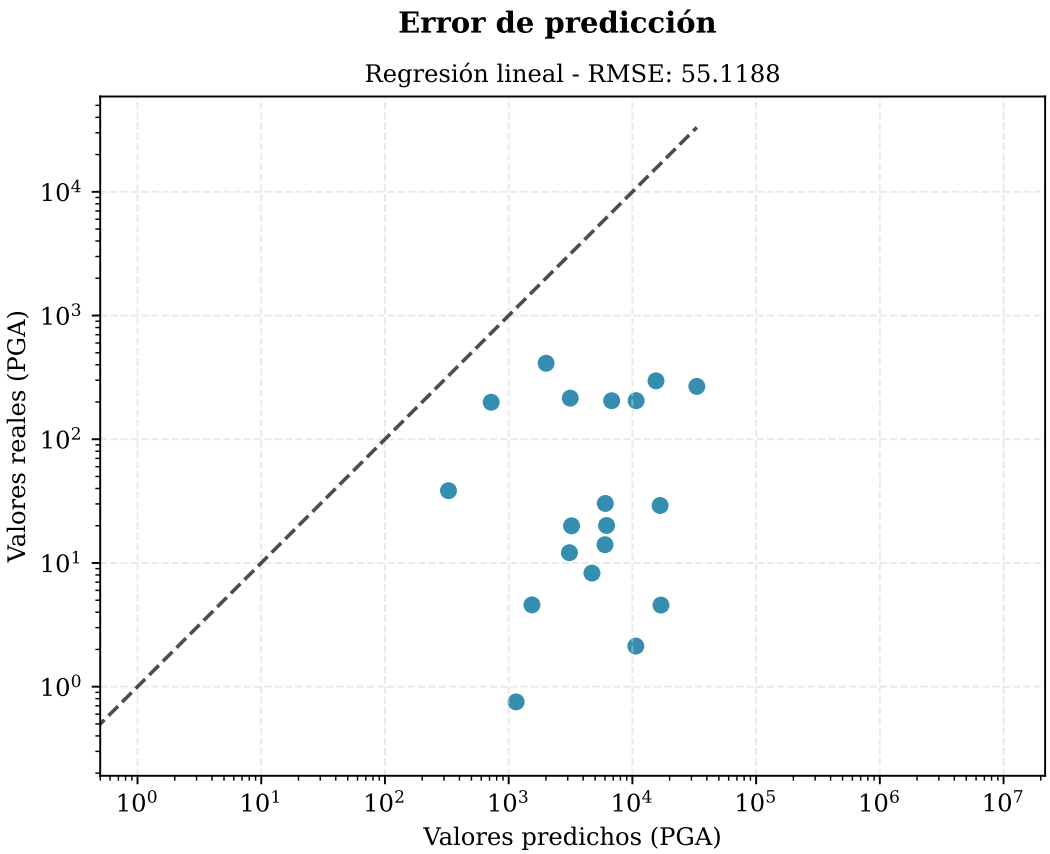

Figura 5.1: Error de predicción con regresión lineal ordinaria.

más robusto a colinealidad, las aproximaciones lineales son incapaces de adaptarse a los datos correctamente.

5.1.3. Red elástica lineal

Para complementar los modelos anteriores, el modelado por red elástica promete resultados con una mayor robustez y precisión al contar con regularización lineal y cuadrática simultáneamente. Como modelo lineal final, se propone con el fin de que se abarquen los modelos lineales más comunes.

La gráfica de error de predicción correspondiente a la red elástica lineal en la Figura 5.3 muestra que su ajuste sigue el mismo comportamiento que los modelos lineales anteriores, generando una geometría similar. Su valor de RMSE es de 57.05, siendo el mayor de los algoritmos lineales, continúa siendo demasiado alto para considerarse un predictor cuyos resultados se puedan aplicar en problemas reales. De manera similar, su puntuación R^2 de 0.2019 lo convierte en el algoritmo lineal de peor desempeño entre los probados.

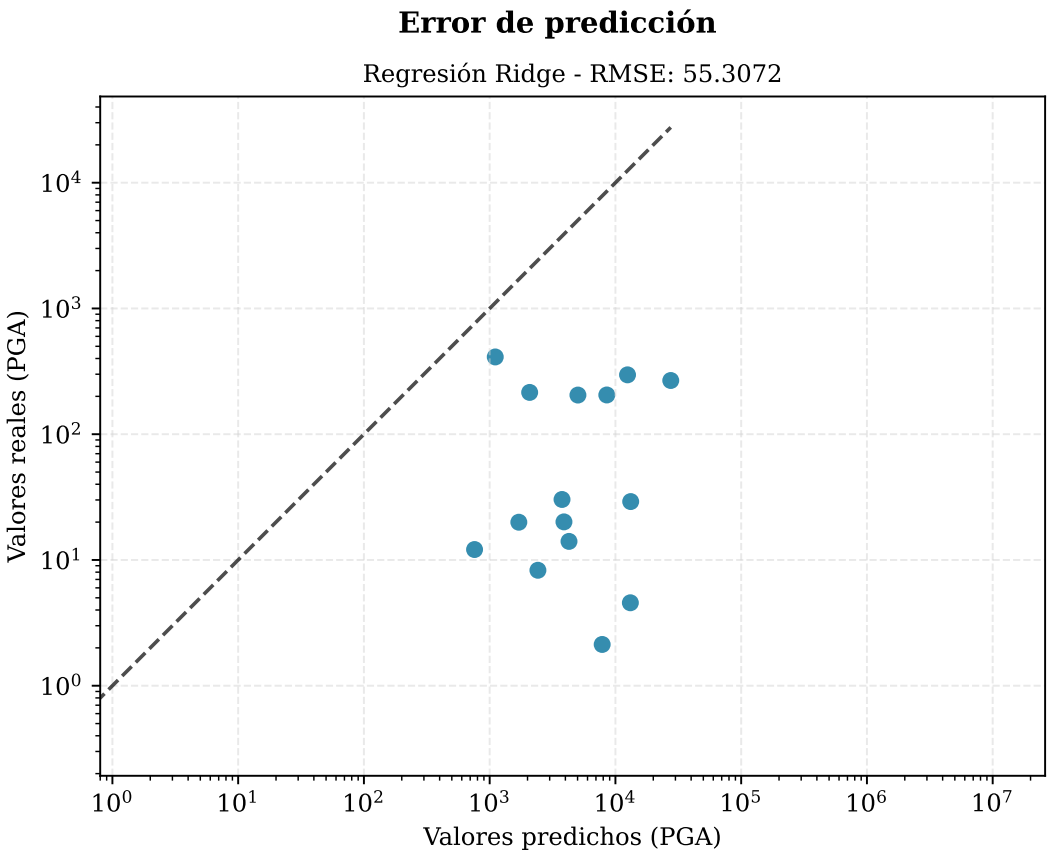

Figura 5.2: Error de predicción con regresión de cresta

De los método lineales se puede concluir, entonces, que sus resultados no son suficientes para generar predicciones fiables en los datos sísmicos utilizados, por lo que no deberían considerarse durante el entrenamiento de modelos basados en datos similares.

5.2. Árboles de decisión

Dado que los árboles de decisión no dependen del ajuste de una función matemática analítica, se espera que ofrezcan una mayor precisión en comparación con los modelos lineales. Además, al tratarse de árboles de decisión simples, su costo computacional es relativamente bajo en comparación con los modelos más complejos que se analizarán posteriormente.

En la Figura 5.4 se observa la gráfica de error de predicción del modelo hecho a partir del entrenamiento de árboles de decisión simples. Con un error cuadrático medio de 35.79, este modelo ya muestra un desempeño mucho mejor que el de los modelos lineales, presentando una dispersión de puntos muy cercanos a la línea punteada. Este árbol de

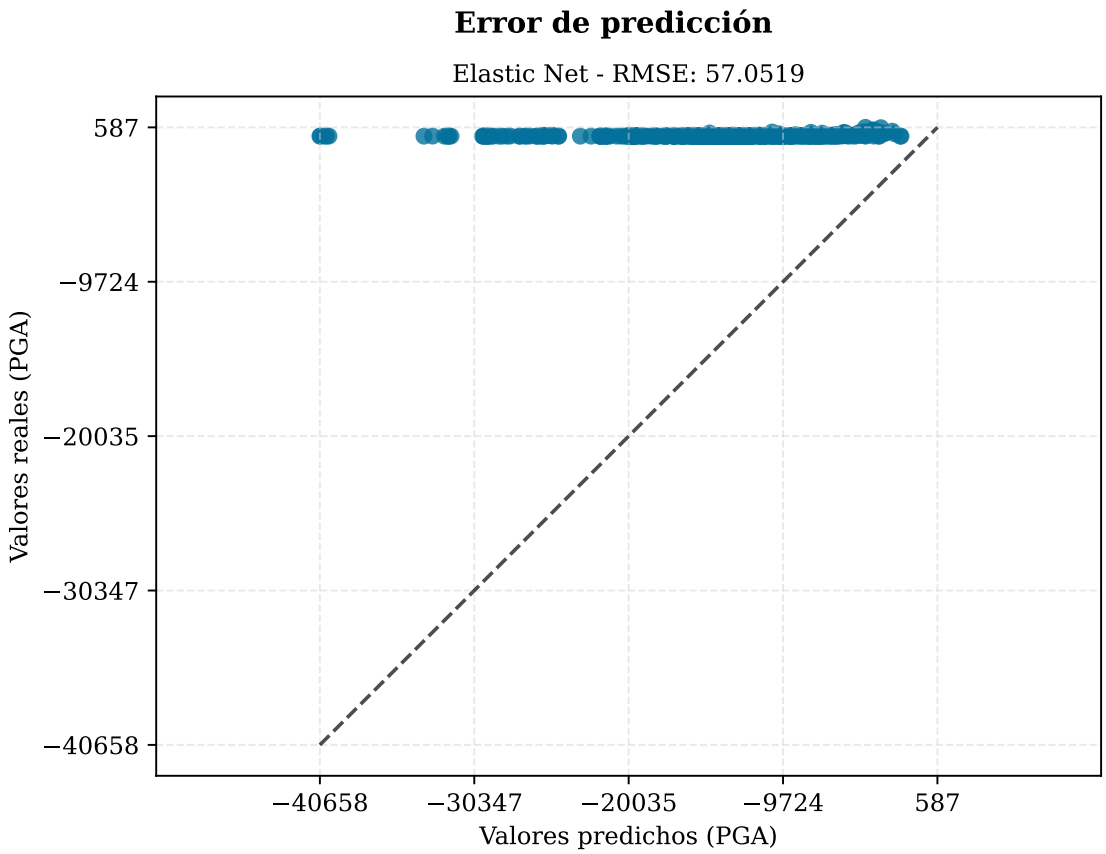

Figura 5.3: Error de predicción con regresión Elastic Net

decisión tiene una puntuación R^2 de 0.6859 y un RMSE de 35.79, lo que representa un nivel de generalización ante datos no vistos del 68.59 %, que ya podría ser utilizable en predicciones reales de este tipo de datos. Si bien el modelo está lejos de ser óptimo, muestra una capacidad de ajuste a los datos muy superior a los modelos paramétricos.

5.3. Métodos de conjunto

Los métodos de conjuntos, que combinan múltiples estimadores para mejorar la capacidad de generalización, se presentan como fuertes candidatos para ser el modelo óptimo. Aunque tienen el costo computacional más alto entre todos los modelos evaluados, también podrían ofrecer la mayor precisión. Por ello, se probarán algunos de los métodos más comunes para analizar su desempeño.

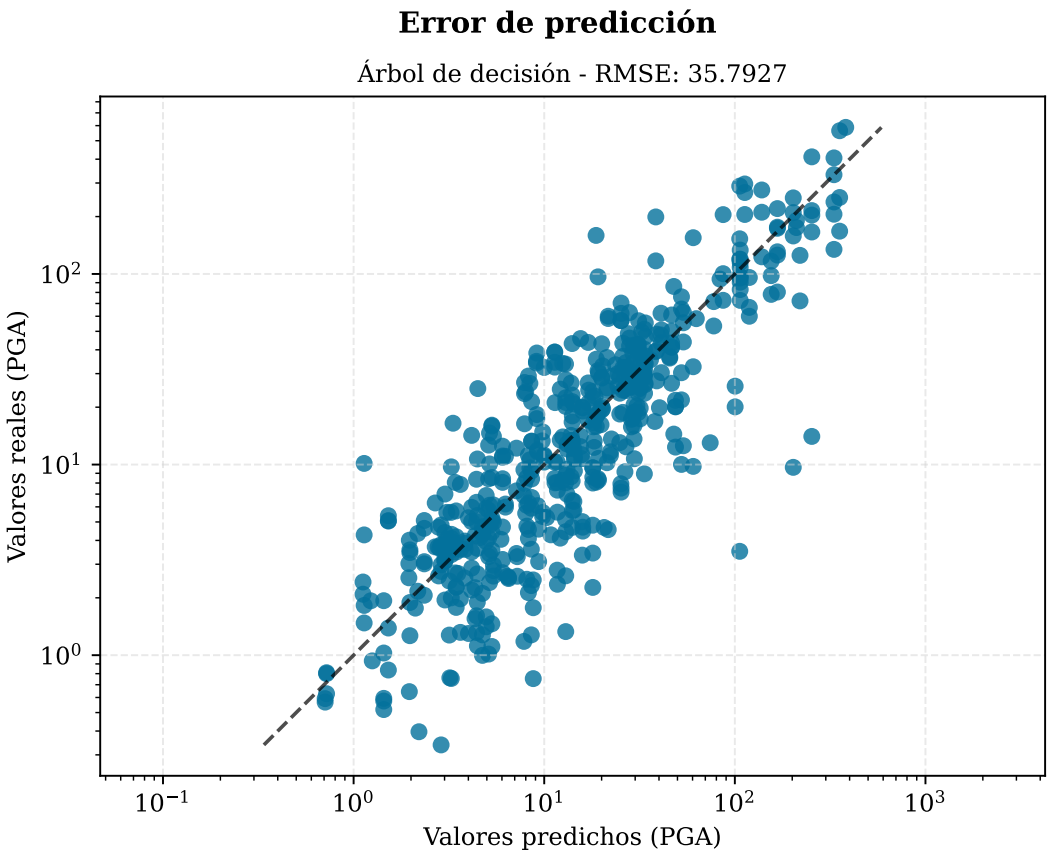

Figura 5.4: Error de predicción de árboles de decisión

5.3.1. Bosques aleatorios

Los bosques aleatorios suelen adaptarse bien a distintos tipos de datos y ofrecer una alta precisión. Sin embargo, tienen uno de los costos computacionales más elevados para la búsqueda de hiperparámetros y el entrenamiento. Aun así, se consideran debido a ser uno de los métodos compuestos más comunes, con un alto potencial para ofrecer resultados óptimos.

En la Figura 5.5 se observa la gráfica de error de predicción en el entrenamiento de bosques aleatorios. Este modelo presenta una dispersión mayor en los valores más pequeños, ocasionando que aquellos valores por debajo de 5 gales no puedan ser vistos por el algoritmo. Esto puede deberse a que los árboles creados durante el entrenamiento no hayan sido capaces de alcanzar a adaptarse a estos valores debido a los parámetros de creación de los árboles durante su optimización. Sin embargo, el bosque aleatorio cuenta con una precisión muy ligeramente superior respecto a los árboles de decisión simples en los valores mayores, por lo que su valor de RMSE y puntuación R^2 son más altos (34.78 y 0.7034, respectivamente).

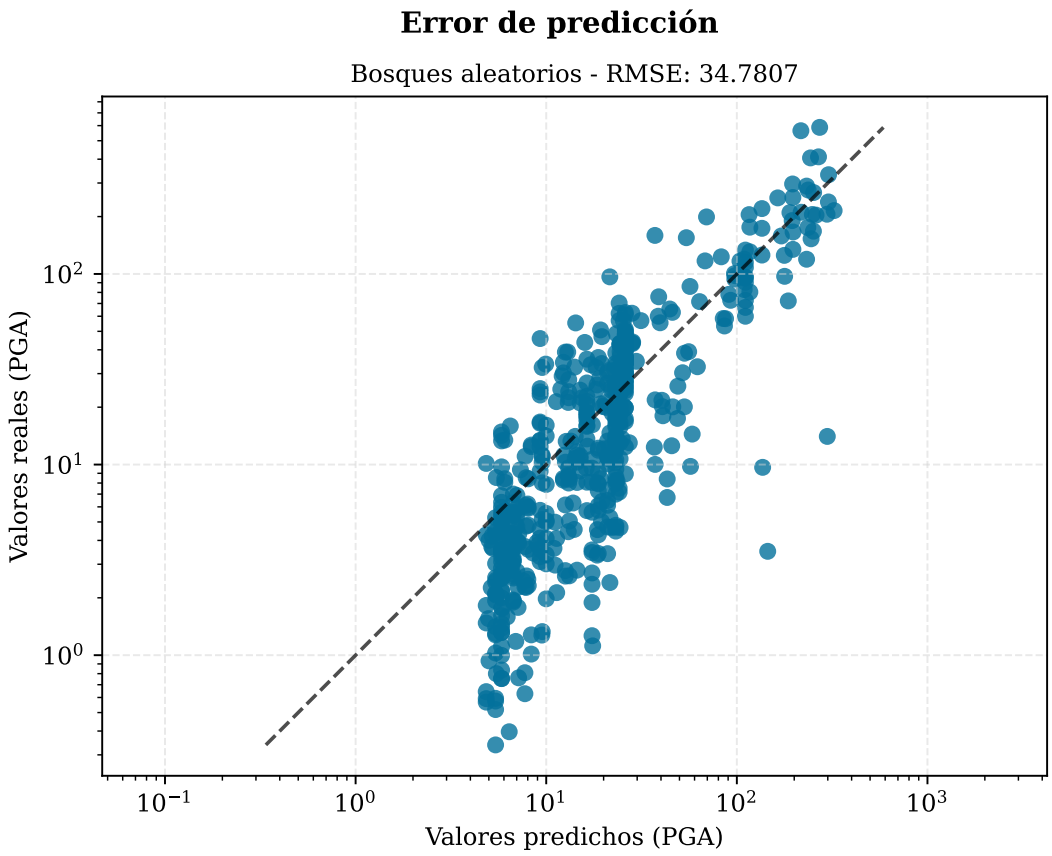

Figura 5.5: Error de predicción de bosques aleatorios

5.3.2. Impulso de gradiente basado en histograma

Dado que este método promete disminuir órdenes de magnitud el tiempo de ajuste cuando el conjunto de datos tiene más de una decena de miles de observaciones respecto a otros métodos compuestos (Pedregosa et al., 2011), es de especial interés debido a que el conjunto de datos utilizado tiene alrededor de 19 mil muestras, por lo que se espera que se adapte de manera óptima a los datos en un tiempo menor a los bosques aleatorios.

Después de la optimización y entrenamiento, el estimador de impulso de gradiente basado en histograma tiene una puntuación R^2 de 0.7572 y un RMSE de 31.4667, siendo una puntuación ligeramente superior al observado por en los bosques aleatorios. Además, como se observa en la Figura 5.6, tiene una mucho mejor predicción de todo el rango de datos existentes, captando las relaciones de manera precisa en toda la base de datos. Este modelo ofrece uno de las mejores puntuaciones y robustez ante los datos presentados, lo que lo convierte en un candidato muy fuerte para convertirse en el mejor algoritmo.

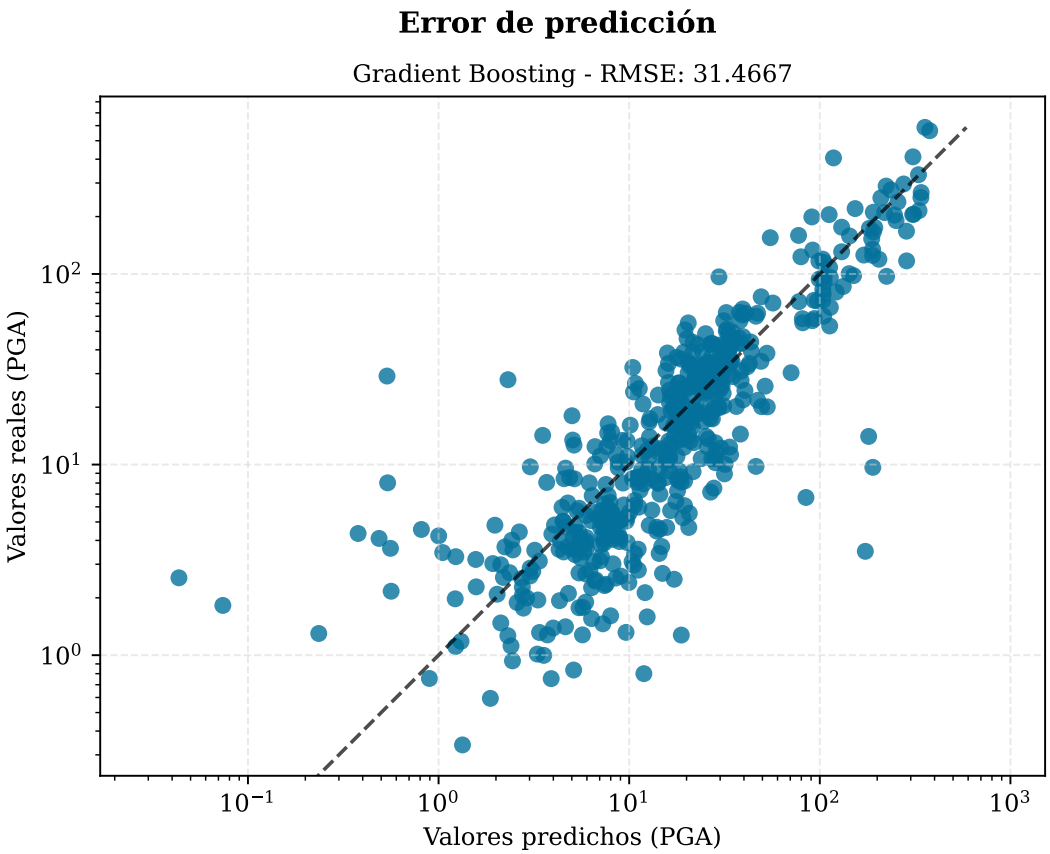

Figura 5.6: Error de predicción de impulso de gradiente basado en histograma

5.3.3. Impulso de gradiente extremo

El impulso de gradiente extremo proporciona una metodología comparable a la de los bosques aleatorios, pero con técnicas de reducción de errores más rigurosas. Se propone como el método de conjunto final para abarcar de manera integral la variedad de estimadores compuestos disponibles. Se espera que este método ofrezca las predicciones más precisas debido a su enfoque iterativo.

Como se observa en la Figura 5.7, su nivel de precisión es el más alto hasta el momento, logrando un RMSE de 31.4024 y una puntuación R^2 de 0.7582, que es marginalmente mejor que el impulso de gradiente basado en histograma. Esta precisión hace que este estimador sea altamente adaptable a los datos presentados, captando la mayor parte de la estructura y pudiendo generar predicciones con un alto nivel de confianza. La puntuación R^2 obtenida indica que el 75.8 % de las predicciones hechas ante datos no vistos son confiables.

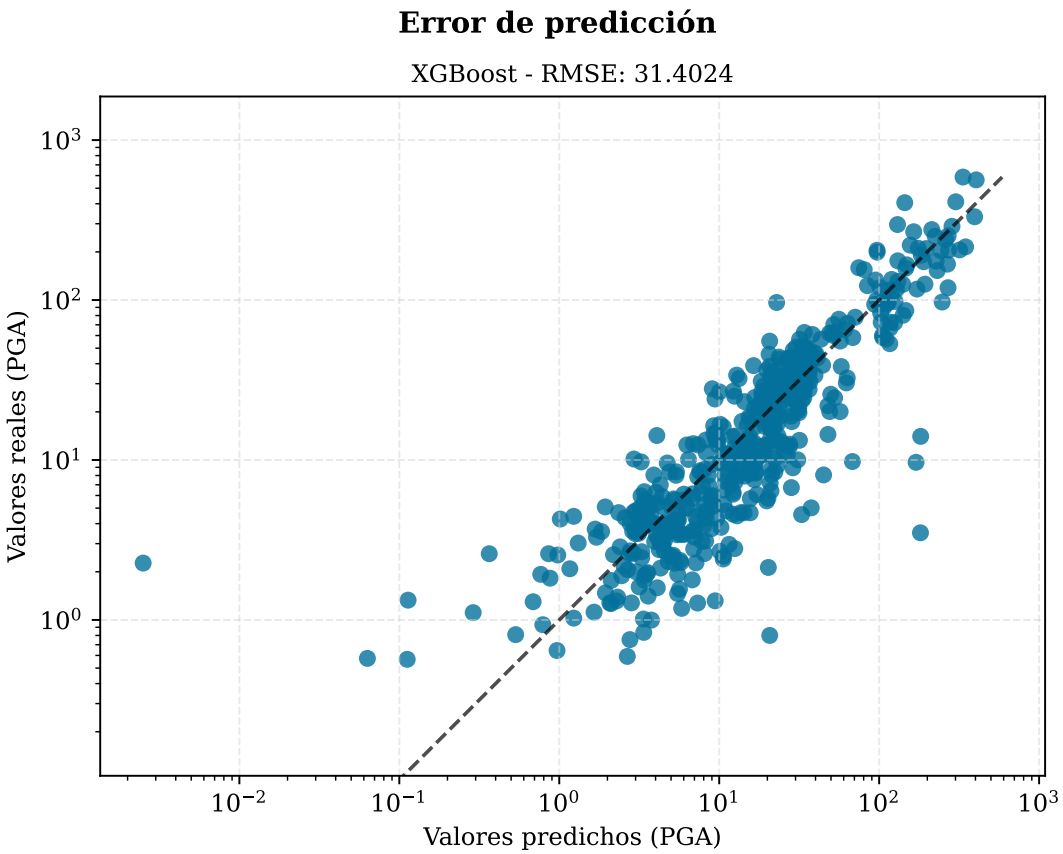

Figura 5.7: Error de predicción de bosques aleatorios

5.4. Reducción de dimensionalidad

Los algoritmos de análisis de reducción dimensionalidad se presentan como una opción para analizar si los modelos con mejor puntuación hasta el momento se pueden beneficiar de un set de datos reducido mediante machine learning supervisado y no supervisado (Mínimos Cuadrados Parciales y Análisis de Componentes Principales, respectivamente). Además de probar el comportamiento de los modelos óptimos ante datos reducidos, se esperan resultados nuevos del algoritmo de Mínimos Cuadrados Parciales, el cual promete una reducción de datos más precisa al ser un método supervisado (Abdi, 2003).

5.4.1. Análisis de componentes principales

Como se mencionó en la Sección 2.5.4, si bien PCA es una herramienta útil para reducir la dimensionalidad con fines de visualización, no es capaz de realizar tareas de regresión por sí misma. Debido a esto, se combina con el estimador que haya tenido un mejor desempeño por sí mismo a través de un *pipeline* como se ilustra en la Figura 4.6, siendo este el de impulso de gradiente extremo, de manera que se analice el comportamiento

de un estimador robusto al entrenarse con datos reducidos de manera no supervisada.

Los resultados obtenidos indican que la aplicación de una reducción de dimensionalidad mediante PCA a un conjunto de datos no lineal no genera un ajuste adecuado. Esto se refleja en un coeficiente de determinación bajo y un tiempo de ajuste significativamente mayor en comparación con otros modelos más precisos, además de un error cuadrático medio considerablemente alto.

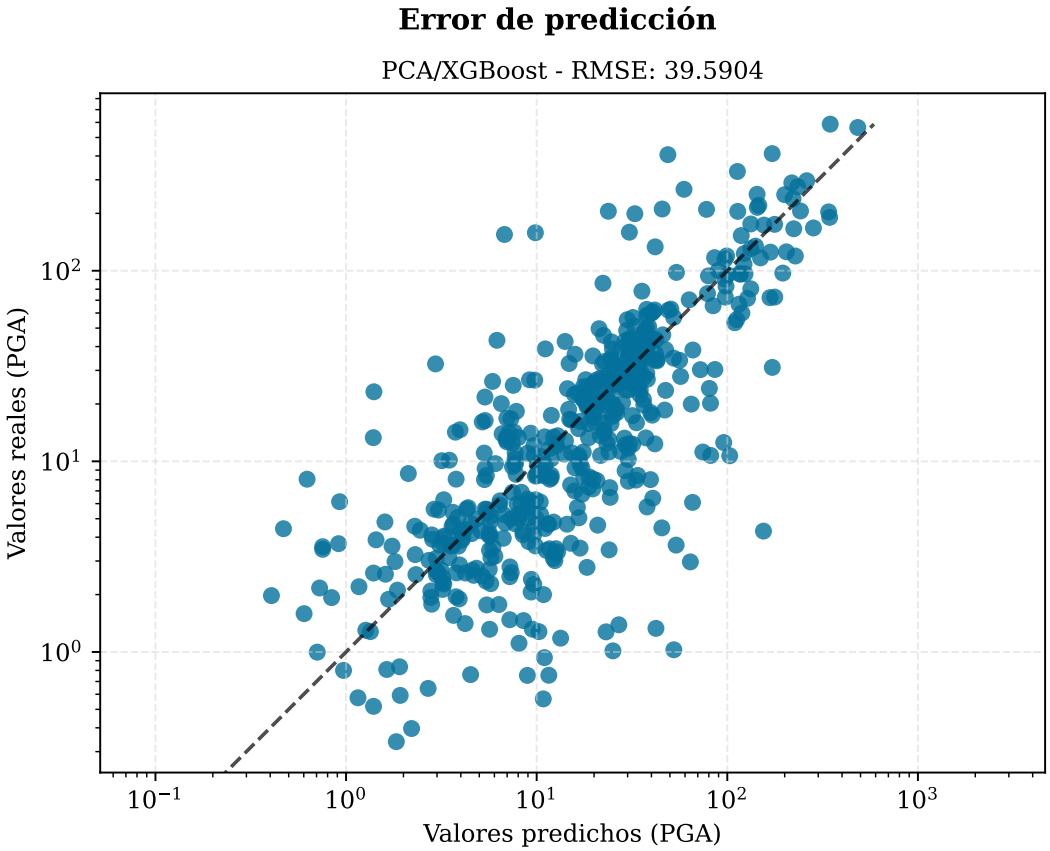

Figura 5.8: PCA con regresión de cresta

En la Figura 5.8 se muestra el error de predicción para el modelo reducido con PCA y predicción con XGBoost. Se observa que, si bien los puntos caen alrededor de la línea central, se encuentran muy separados de esta en forma general, lo que indica que una reducción de dimensionalidad no es el proceso óptimo con este conjunto de datos.

El modelo tiene un RMSE de 39.59 y una puntuación R^2 de 0.6157, colocándose por debajo de los árboles de decisión simples pero considerablemente por encima de los modelos lineales, principalmente gracias al uso de XGBoost como estimador principal.

5.4.2. Mínimos cuadrados parciales

El método de mínimos cuadrados parciales es un algoritmo similar a PCA, pero incluye un proceso de regresión en sí mismo, por lo que no requiere del uso de ningún otro estimador como el caso de PCA. Se elige a manera de comparación con otros algoritmos de reducción de dimensionalidad como PCA.

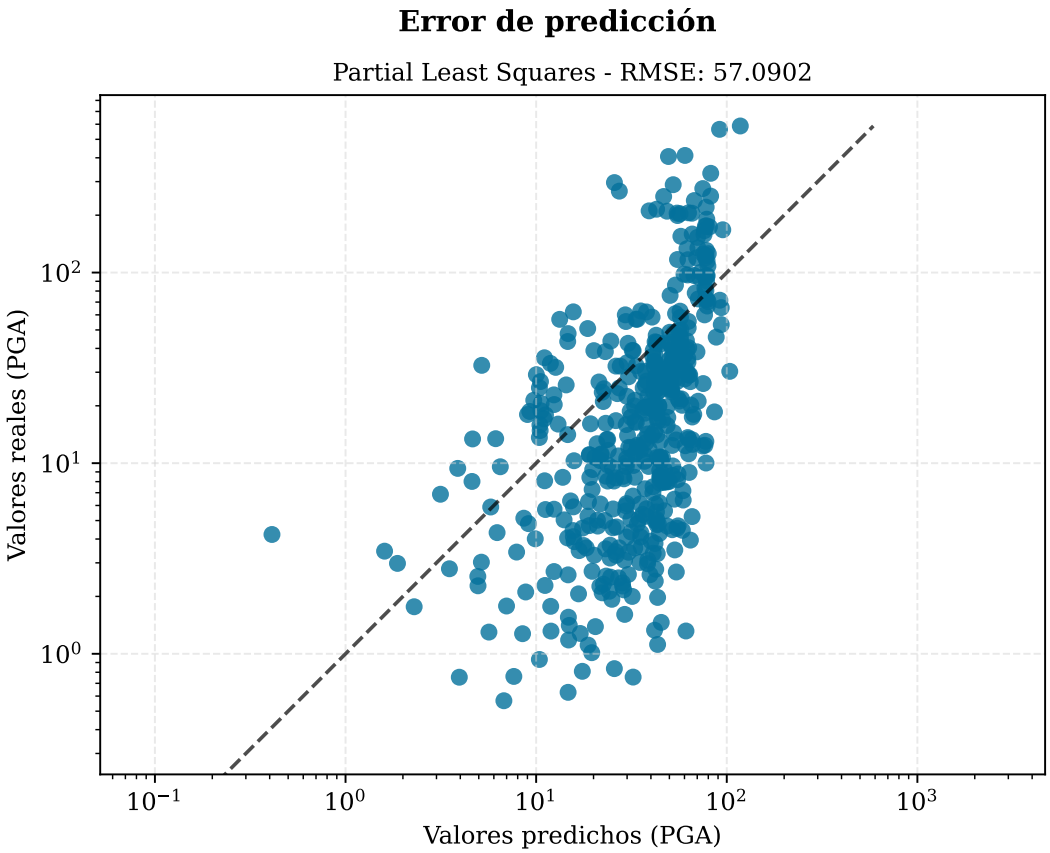

Figura 5.9: Error de predicción de PLS

El modelo de mínimos cuadrados parciales presenta un RMSE de 57.09 y una puntuación R^2 de 0.2008, colocándolo en el último lugar en cuanto a nivel de error entre todos los algoritmos. La razón de esto se puede observar en la Figura 5.9, donde se aprecia que existe una especie de límite de predicción hacia valores de aceleración de 100 gales. Recordando que la gráfica se encuentra en escala logarítmica, los errores crecen en gran medida debido a su incapacidad de predecir valores altos, lo que ocasiona un RMSE extremadamente elevado y un coeficiente de determinación incluso por debajo de los modelos lineales.

Este algoritmo indica que el nivel de adaptación a los datos no es tan limitada como otros modelos. Sin embargo, su RMSE y R^2 se debe a su falla en la predicción de valores

elevados, pero los valores inferiores parecen concentrarse de forma cercana sobre la línea central, a diferencia de lo observado en los modelos lineales (Figuras 5.1, 5.2, 5.3).

Tabla 5.1: Métricas de desempeño de los modelos entrenados

Modelo	**R2**	**RMSE**	**MSE**	**MAE**	**F**
Partial Least Squares	0.2008	57.0902	3259.2922	32.0454	0.5749
Elastic Net	0.2019	57.0519	3254.9222	31.8458	0.6120
Regresión Ridge	0.2499	55.3072	3058.8895	31.1442	0.8082
Regresión lineal	0.2550	55.1188	3038.0793	31.0748	0.0505
PCA - Best	0.6157	39.5904	1567.4028	17.3446	0.0248
Árbol de decision simple	0.6859	35.7927	1281.1147	15.4353	1.0000
Bosques aleatorios	0.7034	34.7807	1209.6944	15.6072	0.1700
Gradient Boosting	0.7572	31.4667	990.1543	13.8604	0.1947
Extreme Boosting	0.7582	31.4024	986.1086	13.0936	0.0122

Al analizar el coeficiente R^2 de forma independiente en la Tabla 5.1, se observa que el modelo de impulso de gradiente extremo alcanza la puntuación más alta, con un valor de 0.7582. Esta puntuación equivale a una precisión del 75.82 % en la predicción de datos no vistos, lo que convierte al modelo XGB en el mejor candidato en términos de la precisión del ajuste del modelo. Tomando como variable principal el error cuadrático medio, el estimador XGB también contiene el mejor valor, con 31.4024.

En contraste con los modelos lineales, los modelos basados en árboles son métodos no lineales, los cuales emplean decisiones jerárquicas o conjuntos de estas para aproximarse a los datos. Esta característica les permite adaptarse de forma más flexible a relaciones complejas, incluso cuando la correlación matemática no es evidente. Como consecuencia, estos modelos alcanzan puntuaciones R^2 significativamente más altas y errores más bajos.

Los resultados finales del desempeño de los modelos en términos de su coeficiente de determinación se encuentra en la Figura 5.10, ordenados de mayor a menor. Se observa que los algoritmos basados en árboles de decisión tienen las puntuaciones más altas por un margen muy alto, mientras que los basados en transformaciones o ajustes matemáticos (métodos de descomposición y regresiones lineales) tienen un puntuaciones mucho más bajas. Esto ofrece una visión de la complejidad presente en los datos sísmicos utilizados en este trabajo.

Además del coeficiente de determinación, el tiempo de entrenamiento de los algoritmos también es un parámetro importante en términos de eficiencia computacional, de

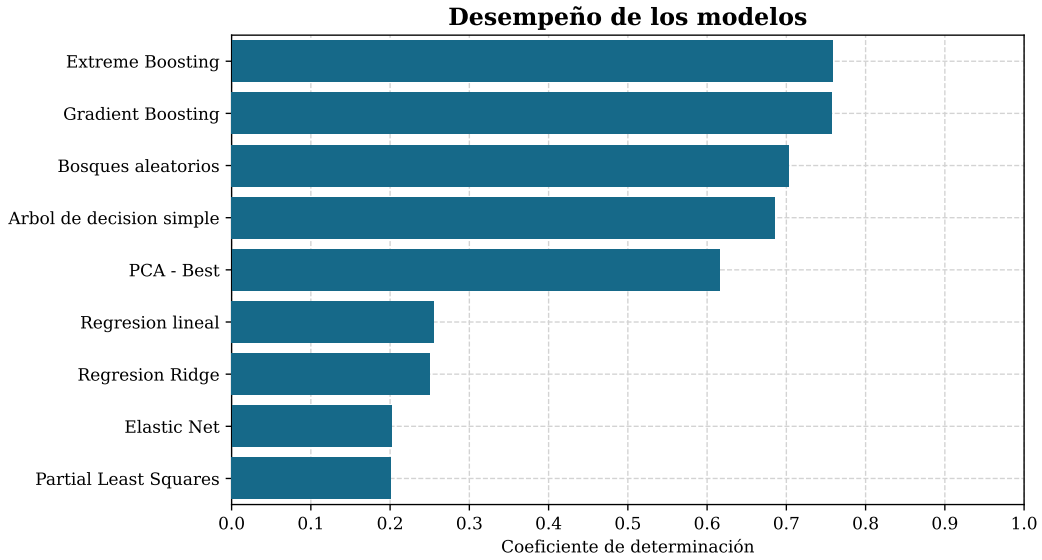

Figura 5.10: Coeficiente de determinación de cada modelo entrenado. Mientras mayor sea la puntuación R^2, mejor generalización tiene el modelo y es capaz de predecir de manera más precisa los datos no vistos.

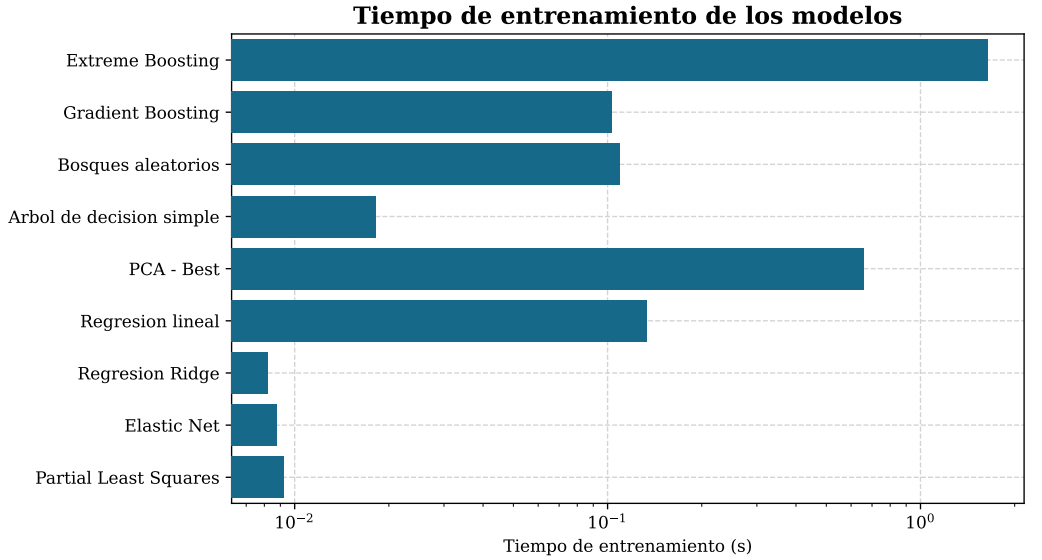

Figura 5.11: Tiempo de entrenamiento de cada modelo con el conjunto de datos completo. Un menor tiempo indica una convergencia más rápida, pero no necesariamente indica una mejor precisión en la predicción.

manera que un algoritmo puede descartarse si su tiempo de entrenamiento es muy alto aunque su puntuación sea la mejor. Para mostrar los tiempos de ajuste para los datos de entrenamiento se muestra la gráfica de la Figura 5.11, con los algoritmos ordenados por puntuación indicando su tiempo de ajuste en segundos. Se puede observar que, si bien los

algoritmos de árboles de decisión son los más precisos de forma absoluta, también cuentan con el costo computacional más alto. No obstante, nótese que el tiempo máximo en todos los algoritmos es de solamente 1 segundo, y de 0.3 segundos para el mejor modelo, por lo que puede no representar una preocupación al momento de elegir el óptimo.

Si se requiere conocer el algoritmo óptimo en cuanto a su relación de precisión-tiempo, se presenta la columna F en la Tabla 5.1, la cual indica en una escala de 0 a 1 un factor que toma en cuenta estos dos elementos, relacionándolos según la siguiente ecuación:

$$F = \frac{R^2}{t} \tag{5.1}$$

El factor F se encuentra normalizado al valor máximo, de manera que represente la eficiencia de cada algoritmo la misma escala. Observando los valores de F, se aprecia que el mejor algoritmo absoluto es el árbol de decisión simple, con una puntuación R^2 de 0.6859 y un tiempo de entrenamiento de solamente 0.018 segundos, ofreciendo una precisión relativamente alta en un tiempo muy bajo.

5.5. Resultados del algoritmo óptimo

Para visualizar de manera más intuitiva los resultados de los modelos entrenados, un ejercicio propuesto es el de predecir el PGA para un determinado sismo a lo largo del área cubierta por los sensores sísmicos disponibles. Con el objetivo de lograr una predicción más precisa, se eligió el evento con el mayor número de registros en la base de datos, el cual corresponde al del 16 de febrero de 2018, con magnitud Mw=7.2. Este sismo tiene registros en 138 estaciones, cubriendo un área de aproximadamente 398,000 km^2. Las mallas son producto de una interpolación Kriging con el software Surfer 13.

En la Figura 5.12 se puede apreciar la distribución de densidad de los datos de PGA utilizados para la predicción del sismo del 2018. Debido al gran rango de valores de PGA, se optó por utilizar una escala logarítmica para visualizar mejor los datos. En esta imagen se puede apreciar que ambos conjuntos de datos tienen una distribución similar, pero se observa que los valores predichos tienden a subestimar la cantidad de valores pequeños de PGA (valores por debajo de 10 cm/s^2) y se concentran más hacia los valores representativos de la media. Este comportamiento puede deberse a la poca cantidad de datos disponibles con esas características durante el entrenamiento, por lo que el modelo es incapaz de predecirlo correctamente.

Debido a que 75 de las 138 estaciones que contienen registros para este sismo se

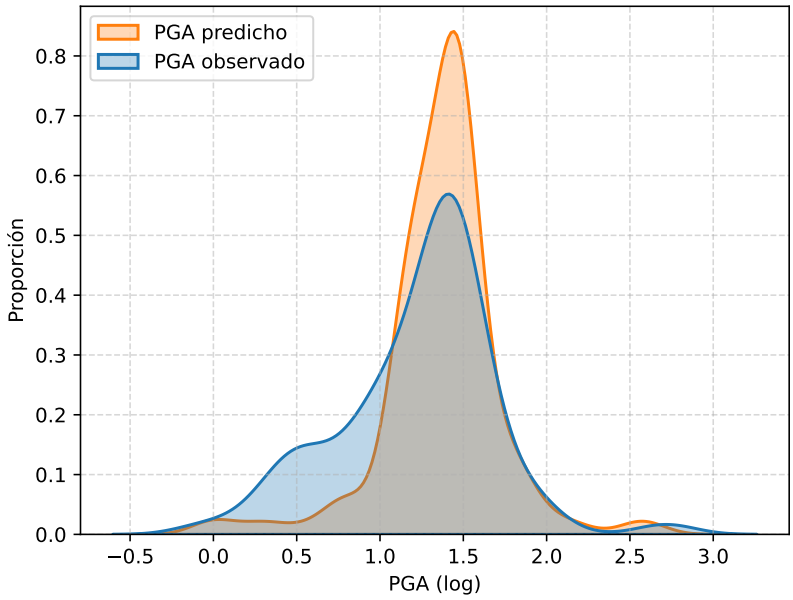

Figura 5.12: Distribución de los datos de PGA predichos con el modelo de impulso de gradiente extremo, comparados con los datos de PGA observados en todo el conjunto de datos.

encuentran en el Valle de México, se realizaron dos mallas de predicción, uno para las estaciones dentro del Valle de México y otro para el resto de las estaciones localizadas a lo largo de la República.

La predicción de valores de PGA de esta región se encuentra en la Figura 5.13, la cual contiene un mapa a la izquierda con la interpolación de los valores reales, mientras que el mapa a la derecha tiene la interpolación de los valores predichos. Se añade como apoyo un mapa base de la zonificación sísmica de la Ciudad de México (UNAM, 2020), pues se puede observar muy claramente cómo los valores más altos de PGA corresponden a la Zona III (zona de lago), mientras que los valores de PGA se reducen de manera gradual a medida que se acercan a la Zona I y Zona II (zonas de lomas y transición, respectivamente).

Si bien la estructura y forma general de los datos predichos por el modelo se mantienen similares a los datos observados, se puede apreciar que existe una subestimación de valores de PGA pequeños, de manera que los valores presentes en las regiones correspondientes a la zona de transición se mantienen hacia la zona de lomas, contrario a lo que se observa en los datos reales, donde este valor disminuye a medida que se entra a la zona de lomas. De manera similar, el modelo es incapaz de predecir con precisión zonas de valores bajos aislados, como los presentes en las zonas centro, superior e izquierda del mapa de valores originales.

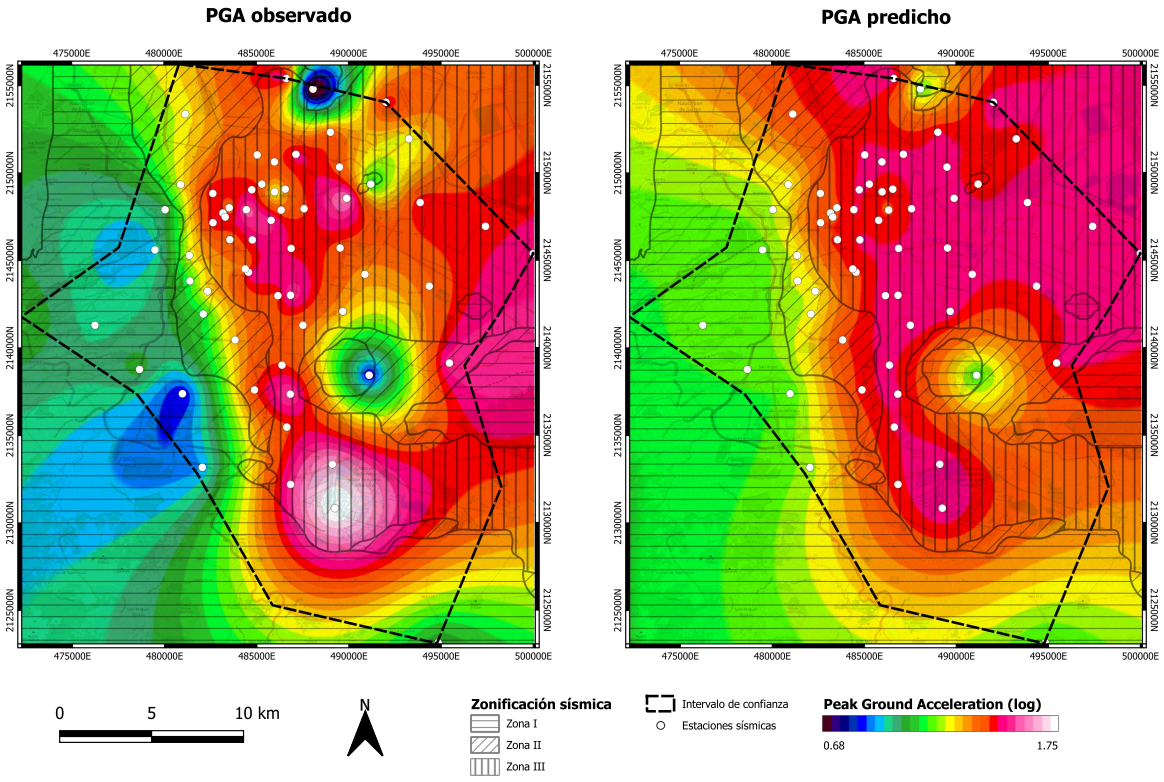

Figura 5.13: Aceleración máxima registrada por las estaciones en el Valle de México para el sismo del 16 de febrero de 2018 Mw=7.2. Ambos mapas comparten la misma escala de color. Nótese la relación con la zonificación sísmica

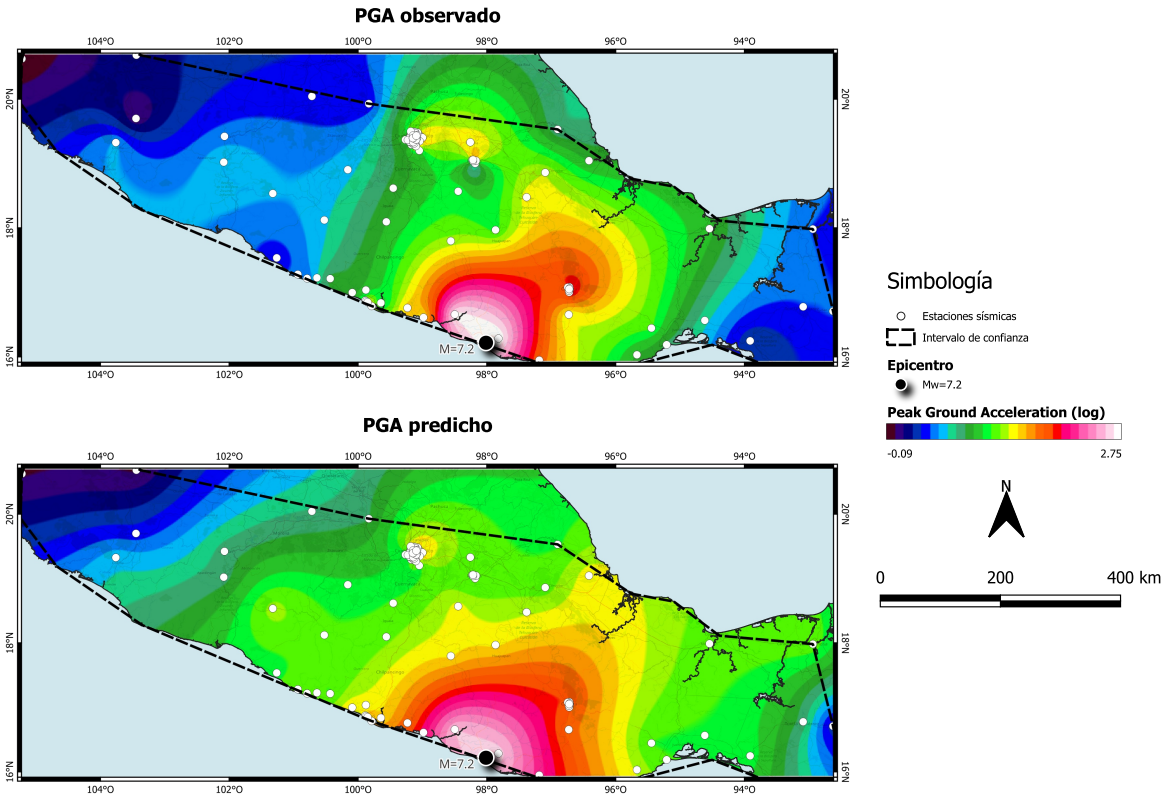

Figura 5.14: Aceleración máxima registrada por las estaciones en México para el sismo del 16 de febrero de 2018 Mw=7.2. Ambos mapas comparten la misma escala de color.

Por otra parte, en la Figura 5.14 se observa la interpolación de valores de PGA para las estaciones del mismo terremoto localizadas fuera del Valle de México en la zona sur de la República. En la parte superior se encuentra el mapa con los valores de PGA reales, mientras que en la parte inferior se encuentra el mapa con los valores predichos.

En el mapa regional de la Figura 5.14 se aprecia que la variación de valores entre ambos mapas es mínima en las zonas más cercanas al epicentro (marcado con una punto color negro), pero estos valores difieren en mayor medida conforme la distancia aumenta. De manera similar a lo observado en el Valle de México en la Figura 5.13, el modelo engloba las relaciones y estructuras generales, pero carece de la precisión para predecir de manera correcta zonas aisladas o con distribuciones espaciales complicadas, como se observa en la zona superior izquierda o hacia el centro.

El comportamiento del algoritmo al predecir datos sísmicos refleja lo presentado en la Figura 5.12, que indica una mayor concentración de valores mayores y una subestimación de valores menores, lo que inherentemente significa una menor variabilidad de los datos.

Esto se ve en los mapas de las Figuras 5.13 y 5.14 como una distribución suavizada de los valores de PGA respecto a los observados por las estaciones durante el sismo.

5.6. Simulación de sismos hipotéticos

El modelo desarrollado, además de predecir datos conocidos, permite simular la respuesta del suelo en términos de su PGA, ante eventos con parámetros conocidos. Dado que los parámetros necesarios en el modelo son fáciles de simular, permite crear escenarios hipotéticos y realizar una aproximación de la respuesta de sitio en cada una de las estaciones en las que se entrenó el algoritmo.

Se proponen dos eventos a manera de ejemplo para visualizar la precisión y confiabilidad del modelo:

- Sismo interplaca de régimen compresivo con epicentro en las costas de Oaxaca (16.091° N, 98.055° W, 13 km al SE del municipio de Santa María Chicometepec), $M_W = 6.8$, a una profundidad de 14 km.
- Sismo interplaca de régimen compresivo con epicentro en el estado de Michoacán (18.002° N, 103.061° W, 33 km al SE del municipio de Caleta de Campos), $M_W = 7.8$, a una profundidad de 15 km.

Ambos eventos se localizan en la zona de subducción de la Placa Cocos respecto a la Norteamericana, en zonas en las cuales existe una alta incidencia de sismos según los datos históricos, además de encontrarse en las zona analizadas durante el entrenamiento del modelo.

Para simular estos eventos se utiliza el modelo de impulso de gradiente extremo, determinado como mejor en la Tabla 5.1 según su coeficiente de determinación, con la ayuda de una función diseñada para recibir los parámetros del sismo y las coordenadas de los puntos donde se desea predecir (todas las estaciones sísmicas en la base de datos, en este caso), devolviendo una tabla con los valores de PGA predichos en los puntos especificados.

```
1
2 epicentro = [16.091,-98.055]
3 simulate(magnitude=6.8,
4         depth=14,
5         coords=epicentro,
6         where=stations,
7         save=True)
```

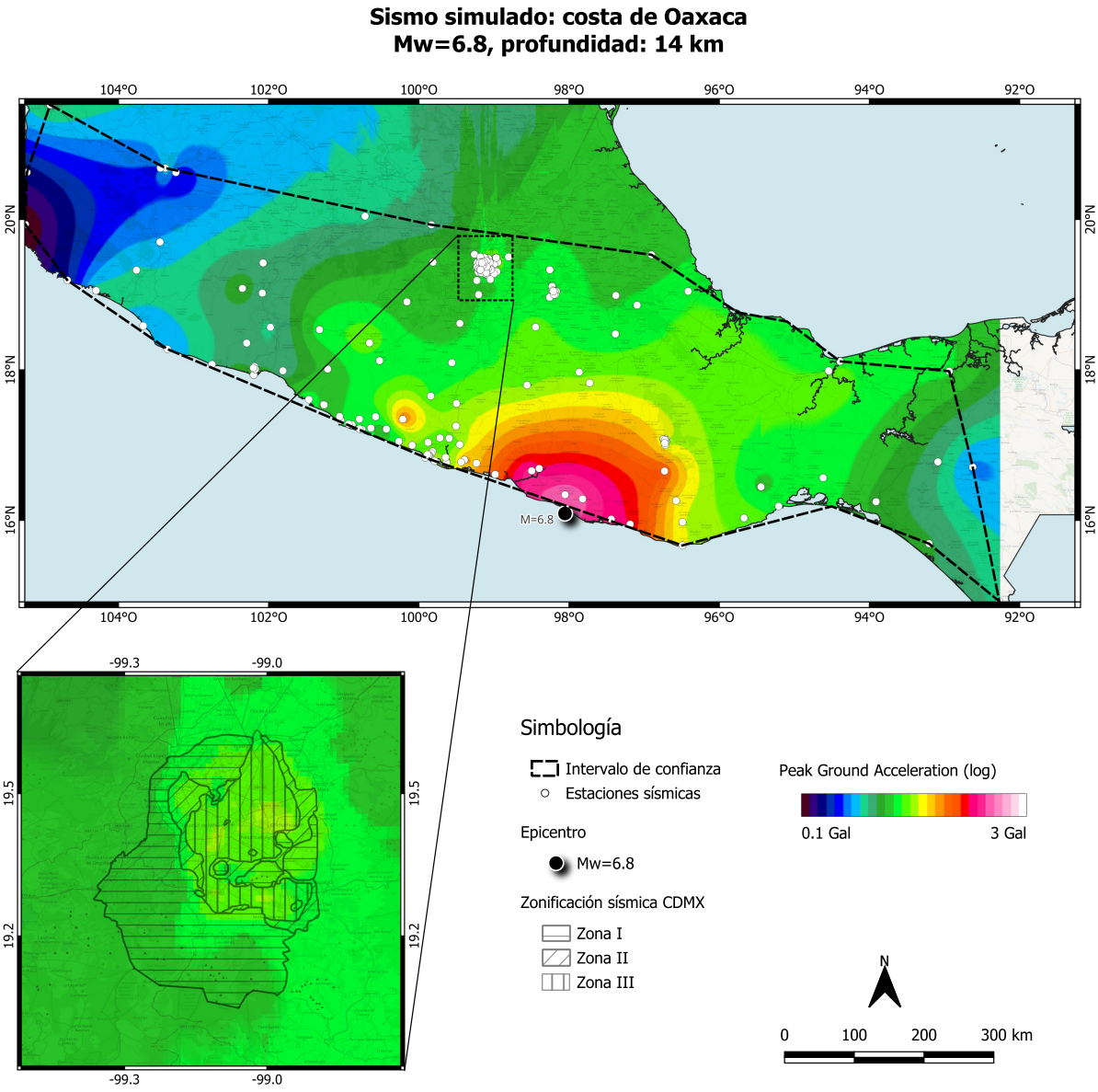

Figura 5.15: Simulación de un sismo $M_W = 6.8$ con epicentro en Michoacán, cuyas aceleraciones fueron estimadas con el modelo XGBoost.

Los valores obtenidos son interpolados mediante el método Kriging con Surfer 13 para obtener un mapa de intensidades sísmicas. En la Figura 5.15 se presentan los resultados del sismo simulado en Oaxaca, de magnitud $M_W = 6.8$. En este mapa se puede observar que la intensidad máxima se registra en las estaciones más cercanas y va disminuyendo conforme aumenta la distancia, y presenta cambios en la intensidad debido a las características del suelo en distintas regiones introducidas en la simulación, siguiendo ligeramente el patrón de intensidades presente en un sismo real de características similares en la Figura 5.14.

Se debe observar el detalle de las estimaciones en la zona del Valle de México, presente en el mapa de la esquina inferior izquierda. Para ayudar a distinguir la región, se sobrepone una capa de la zonificación sísmica. Se aprecia que los valores de PGA dentro de la Zona III (zona de lago) son ligeramente más altos que los de las zonas I y II, lo que indica una buena adaptabilidad a las condiciones del suelo durante la simulación de sismos. La simulación tiene un valor mínimo de 1.29 gales y un máximo de 358.6 gales.

Por otra parte, la Figura 5.16 contiene la simulación de respuesta sísmica ante un sismo en Michoacán con magnitud $M_W = 7.8$, a una profundidad de 15 kilómetros. Al igual que la simulación anterior y conforme al comportamiento esperado, se observa una disminución del PGA a medida que aumenta la distancia del epicentro, pero el efecto de sitio se hace presente en diversas localizaciones. El detalle de la zona del Valle de México muestra de forma mucho más clara que el modelo simula de manera correcta la respuesta de la zona de lago según la zonificación sísmica, donde se simularon valores de aceleración más altos que en las zonas aledañas.

Debe notarse que existe un valor de PGA predominante mucho más alto que el de la Figura 5.15 (recordando que la escala M_W es logarítmica, este último evento representa una energía 10 veces mayor que el primero), con un valor mínimo de 28.257 gales y un máximo de hasta 501.87, lo que indica un nivel de intensidad mucho mayor en todo el país.

Ambos mapas comparten la misma escala de color para facilitar su comparación, además de contar con una línea punteada que indica el intervalo de confianza de los datos, pues más allá del alcance de las estaciones sísmicas solamente son estimaciones del interpolador y no deben tomarse como reales.

En la Figura 5.17 se presenta la distribución de valores para cada una de las simulaciones propuestas en comparación con un evento real. Se aprecia que, efectivamente, el sismo de $M_W = 7.8$ tiene una distribución mucho más asimétrica y angosta, inclinada hacia los valores mayores de PGA, mientras que el sismo de $M_W = 6.8$ tiene una distribución similar a la del sismo real de $M_W = 7.2$, con la diferencia de que este último cuenta

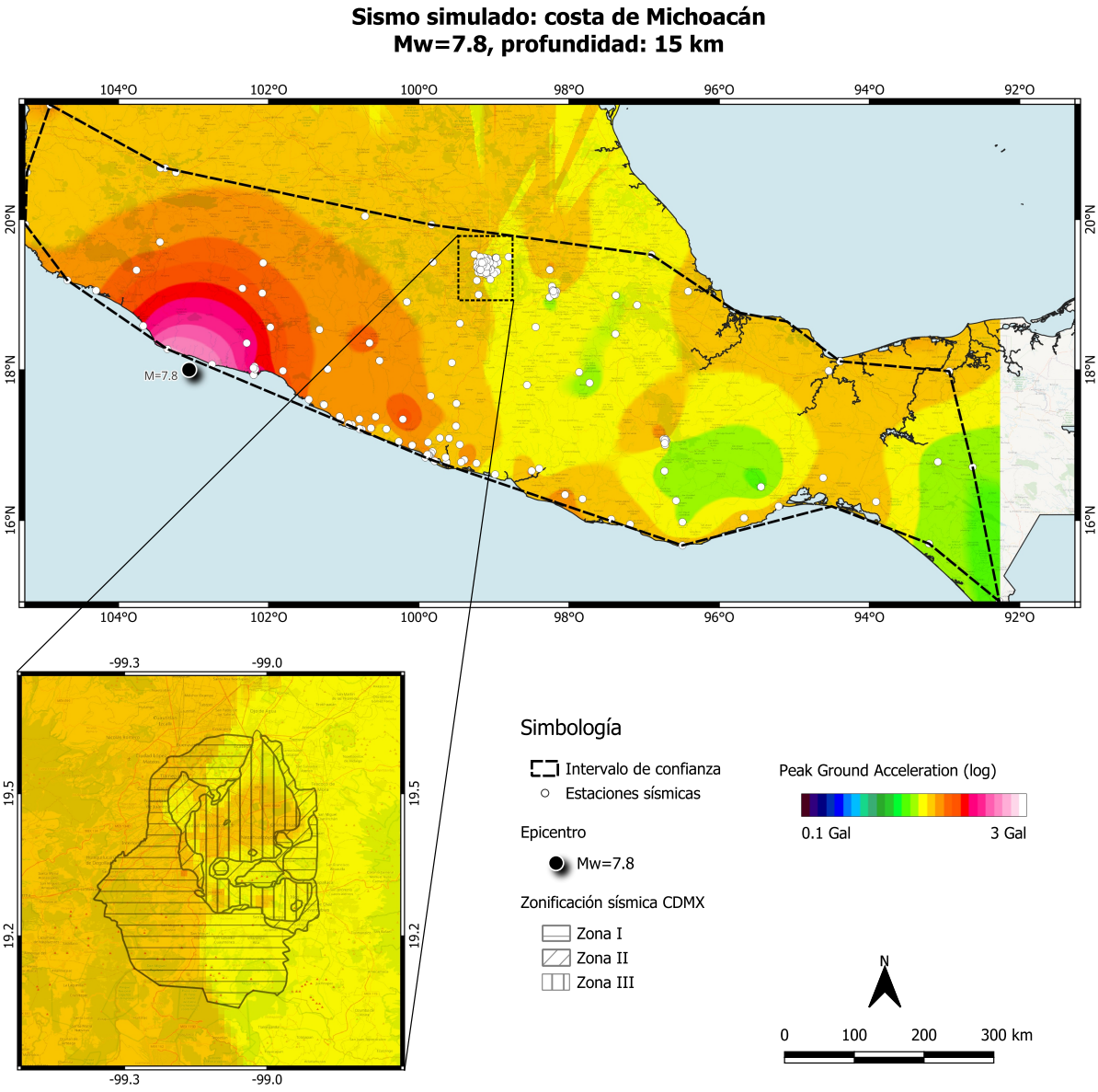

Figura 5.16: Simulación de un sismo $M_W = 7.8$ con epicentro en Michoacán, cuyas aceleraciones fueron estimadas con el modelo XGBoost.

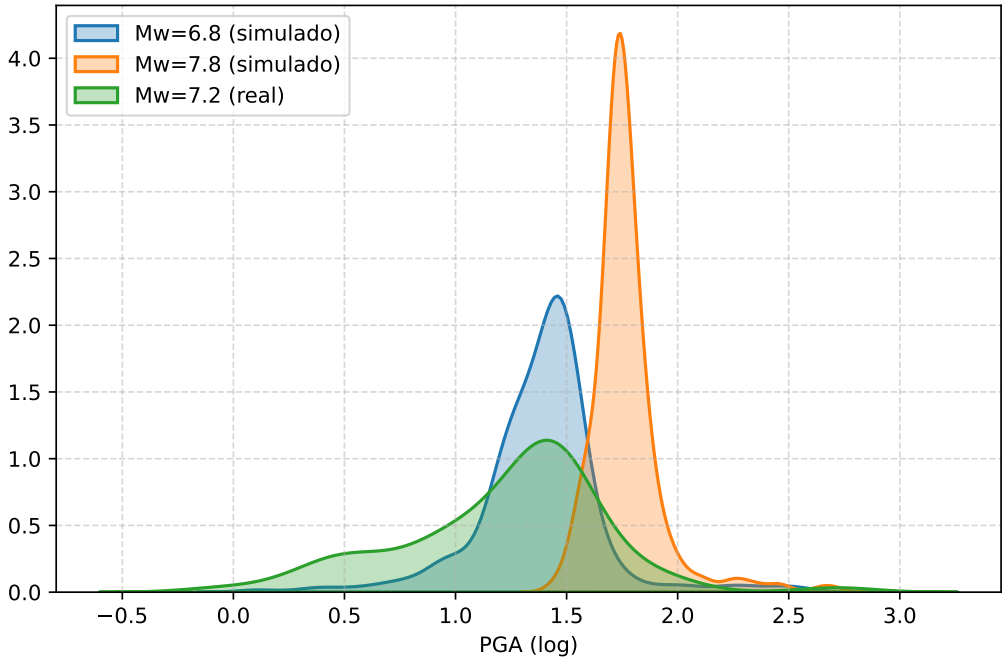

Figura 5.17: Distribución de valores de PGA para las simulaciones de las Figuras 5.15 y 5.16, en comparación con los valores del sismo real en la Figura 5.14.

con una distribución que abarca más valores de PGA, tanto menores como mayores. Esta comparación de distribuciones puede indicar que el modelo no capta de manera completa todas las características necesarias de los datos para generar predicciones similares a las observadas en eventos reales.

Las observaciones hechas en estas distribuciones refuerzan los hallazgos en la Figura 5.7 al tener una alta dispersión en los valores bajos y las diferencias contra los datos observados en las Figuras 5.13 y 5.14 al tener un rango de valores más angosto.

Como observación final, los datos predichos y simulados por el estimador de impulso de gradiente extremo muestran que este algoritmo tiene una alta capacidad de adaptación y un potencial de precisión mucho más alto que el actual si es entrenado con una mayor cantidad de datos de calidad.

Capítulo 6

Conclusiones y trabajo futuro

En esta tesis se ha llevado a cabo un análisis exhaustivo de los algoritmos de machine learning más comunes aplicados a la predicción de aceleraciones sísmicas máximas en México. A lo largo del estudio, se ha demostrado que los métodos basados en machine learning pueden superar en practicidad a las técnicas tradicionales de predicción, especialmente en contextos donde los datos sísmicos presentan alta complejidad y variabilidad, y la definición de modelos matemáticos se torna inviable.

Entre los hallazgos más significativos, se destaca que el algoritmo de impulso de gradiente extremo (XGBoost) ha mostrado un desempeño superior con una precisión del 75.8 % (nótese la eficacia de este algoritmo, relativamente sencillo, contra investigaciones similares, donde se obtienen precisiones del 77 % utilizando arquitecturas de algoritmos significativamente más complejas (Joshi et al., 2024)), superando a otros modelos lineales y de árboles de decisión tanto simples como compuestos. El modelo XGBoost es capaz de simular eventos hipotéticos de manera precisa y lógica, únicamente estando limitado por la cantidad de datos en los que fue entrenado. Además, el uso de hardware especializado para machine learning, como unidades de procesamiento gráfico (GPU) o unidades de procesamiento de tensores (TPU), podría optimizar la selección de hiperparámetros, llevando a precisiones aún más altas en menos tiempo. Esto sugiere que el desempeño de XGBoost tiene margen para mejoras adicionales en investigaciones futuras.

Este estudio también enfatiza la importancia de contar con bases de datos sísmicas de alta calidad y la necesidad de continuar mejorando la infraestructura de registro y monitoreo sísmico en México. La capacidad de predecir con mayor exactitud las aceleraciones sísmicas puede contribuir significativamente a la preparación y respuesta ante eventos sísmicos, reduciendo potencialmente los impactos materiales y humanos.

Para trabajos futuros, se identifican varias áreas de investigación que podrían expandir y mejorar los resultados obtenidos en esta tesis, algunas de las cuales se describen a continuación:

1. **Ampliación de la base de datos**: Incluir datos de eventos sísmicos recientes y de diferentes regiones podría aumentar la robustez y generalizabilidad de los modelos. Además, la inclusión de una red de estaciones más amplia (por ejemplo, del Servicio Sismológico Nacional), podría mejorar en gran medida la precisión del algoritmo.

2. **Exploración de nuevos algoritmos**: Explorar algoritmos de machine learning y deep learning adicionales, como redes neuronales profundas y modelos de aprendizaje reforzado, que puedan capturar de manera más efectiva las relaciones no lineales y complejas en los datos sísmicos, generando así predicciones más precisas.

3. **Inclusión de un mayor número de características**: Incorporar datos adicionales, como información geológica, datos de movimiento del terreno por GPS durante un sismo, mecanismos focales y otras señales o datos relevantes, permitiría enriquecer el conjunto de datos y mejorar la precisión de las predicciones.

4. **Colaboraciones interdisciplinarias**: Explorar colaboraciones con expertos en geología, ingeniería civil, geofísica, ciencias de datos y políticas públicas para abordar de manera integral los desafíos asociados a la predicción y mitigación de riesgos sísmicos.

5. **Desarrollo de modelos en tiempo real**: Una vez desarrollado un modelo lo suficientemente preciso, se pueden procesar y analizar datos en tiempo real, proporcionando alertas y predicciones inmediatas que puedan ser utilizadas por los sistemas de protección civil y alerta temprana.

Este trabajo representa un paso inicial hacia la posible mejora de las predicciones de intensidades sísmicas en México mediante el uso de técnicas avanzadas de machine learning. No obstante, se requiere de una mejora continua para perfeccionar estos modelos y maximizar su impacto, tanto en el ámbito académico como el social y político.

Bibliografía

Abdi, H. (2003). Partial Least Squares (PLS) Regression. *Encyclopedia of Social Sciences Research Methods*.

Alkhalifah, T., Song, C., Waheed, U. B., & Hao, Q. (2021). Wavefield solutions from machine learned functions constrained by the Helmholtz equation. *Artificial Intelligence in Geosciences*, *2*, 11-19.

Birnie, C., Ravasi, M., Liu, S., & Alkhalifah, T. (2021). The potential of self-supervised networks for random noise suppression in seismic data. *Artificial Intelligence in Geosciences*, *2*, 47-59.

Bozorgnia, Y., & Bertero, V. (2004). *Earthquake engineering from engineering seismology to performance-based engineering*. International Code Council.

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5-32.

CENAPRED. (2024a). Aniversario 63 del sismo del Ángel de 1957. https://www.gob.mx/cenapred/articulos/aniversario-63-del-sismo-del-angel-de-1957

CENAPRED. (2024b). Los sismos del 19 septiembre. https://www.gob.mx/cenapred/articulos/los-sismos-del-19-septiembre?idiom=es

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785-794. https://doi.org/10.1145/2939672.2939785

CIRES. (2024). Centro de Instrumentación y Registro Sísmico, A. C. http://www.cires.org.mx/cires_n.php

Esteva, L. (1970). *Regionalización sísmica de México para fines de ingeniería*. Instituto de Ingeniería.

Fariza, I. (2017, septiembre). La cercanía del epicentro impidió que las alarmas anticipasen con suficiente tiempo el terremoto. https://elpais.com/internacional/2017/09/20/mexico/1505867871_954911.html

Flores, R. M. S. (2015, octubre). *Evaluación del peligro sísmico en el municipio de Naucalpan de Juárez, Edo. de México*.

García Arróliga, N. M., Méndez Estrada, K. M., Franco Vargas, E., & Olmedo Santiago, C. (2019). Impacto socioeconómico de los principales desastres ocurridos en la república mexicana. *Centro Nacional de Prevención de Desastres*. https://www.cenapred.unam.mx/es/Publicaciones/archivos/415-IMPACTO_SOCIOECONOMICO_2017.PDF

Gobierno de México. (2024). Los sismos del 19 septiembre. https://www.gob.mx/cenapred/articulos/los-sismos-del-19-septiembre?idiom=es

IBM. (2023, octubre). What is ridge regression? https://www.ibm.com/topics/ridge-regression

IdeI UNAM. (2017). Base de datos de Registros Aelerográficos de la Red Sísmica Mexicana. https://aplicaciones.iingen.unam.mx/AcelerogramasRSM/RedAcelerografica.aspx

Joshi, A., Raman, B., Mohan, C. K., & Cenkeramaddi, L. R. (2024). Application of a new machine learning model to improve earthquake ground motion predictions. *Nat. Hazards (Dordr.)*, *120*(1), 729-753.

Kim, Y., & Nakata, N. (2018). Geophysical inversion versus machine learning in inverse problems. *Lead. Edge*, *37*(12), 894-901.

Kristori, R. (2024, enero). Decision Tree. Clearly Explained! https://python.plainenglish.io/decision-tree-clearly-explained-7c74f40dae9c

Kroese, D. P., Botev, Z. I., Taimre, T., & Vaisman, R. (2019, noviembre). *Data science and machine learning. Mathematical and statistical methods*. Chapman; Hall/CRC.

Li, K., Chen, S., & Hu, G. (2020). Seismic labeled data expansion using variational autoencoders. *Artificial Intelligence in Geosciences*, *1*, 24-30.

McDonald, G. C. (2009). Ridge regression. *Wiley Interdiscip. Rev. Comput. Stat.*, *1*(1), 93-100.

Meng, F., Ren, T., Liu, Z., & Zhong, Z. (2023). Toward earthquake early warning: A convolutional neural network for repaid earthquake magnitude estimation. *Artificial Intelligence in Geosciences*, *4*, 39-46.

Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, *7*, 21.

Nwaila, G. T., Zhang, S. E., Bourdeau, J. E., Ghorbani, Y., & Carranza, E. J. M. (2022). Artificial intelligence-based anomaly detection of the Assen iron deposit in South Africa using remote sensing data from the Landsat-8 Operational Land Imager. *Artificial Intelligence in Geosciences*, *3*, 71-85.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825-2830.

Pro, A. (2016). How xgboost algorithm works. https://pro.arcgis.com/en/pro-app/latest/tool-reference/geoai/how-xgboost-works.htm

Qadrouh, A., Carcione, J., Alajmi, M., & Alyousif, M. (2019). A tutorial on machine learning with geophysical applications. *Bollettino di Geofisica Teorica ed Applicata*, *60*(3).

RAII-UNAM. (2023). Base de datos de Registros Acelerográficos de la Red Sísmica Mexicana. https://aplicaciones.iingen.unam.mx/AcelerogramasRSM/Inicio.aspx

Reiss, R.-D., & Thomas, M. (2007). *Statistical Analysis of Extreme Values: with Applications to Insurance, Finance, Hydrology and Other Fields* (3rd ed.). Birkhäuser Basel.

Rogers, S., & Girolami, M. (2020, junio). *A first course in machine learning* (2.ª ed.). CRC Press.

Rouet-Leduc, B., Hulbert, C., Lubbers, N., Barros, K., Humphreys, C. J., & Johnson, P. A. (2017). Machine learning predicts laboratory earthquakes. *Geophys. Res. Lett.*, *44*(18), 9276-9282.

Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, *3*(3), 210-229. https://doi.org/10.1147/rd.33.0210

Sawires, R., Santoyo, M. A., Peláez, J. A., & Corona Fernández, R. D. (2019). An updated and unified earthquake catalog from 1787 to 2018 for seismic hazard assessment studies in Mexico. *Sci. Data*, *6*(1), 241.

SIAP. (2024). Terremoto, México, 1985. https://www.gob.mx/siap/articulos/terremoto-mexico-1985?idiom=es

SSN. (2015). SSN - Red de estaciones sismológicas. http://www.ssn.unam.mx/acerca-de/estaciones/

Staff, F. (2022, septiembre). Septiembre: un mes estigmático para la actividad sísmica en México. https://www.forbes.com.mx/setiembre-un-mes-estigmatico-para-la-actividad-sismica-en-mexico/

UNAM. (2020). Actualización de la zonificación sísmica de la Ciudad de México y áreas aledañas - parte Norte. *Universidad Nacional Autónoma de México*.

USGS. (2009, abril). What is the "ring of fire"? https://www.usgs.gov/faqs/what-ring-fire

USGS. (2020). Vs30 Models and Data. https://earthquake.usgs.gov/data/vs30/

Verleysen, M., & François, D. (2005). The Curse of Dimensionality in Data Mining and Time Series Prediction. En J. Cabestany, A. Prieto & F. Sandoval (Eds.), *Computational Intelligence and Bioinspired Systems* (pp. 758-770). Springer Berlin Heidelberg.

Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Clim. Res.*, *30*, 79-82.

Wood, D. A. (2021). Enhancing lithofacies machine learning predictions with gamma-ray attributes for boreholes with limited diversity of recorded well logs. *Artificial Intelligence in Geosciences*, *2*, 148-164.

Zhang, S. E., Nwaila, G. T., Bourdeau, J. E., & Ashwal, L. D. (2021). Machine learning-based prediction of trace element concentrations using data from the Karoo large igneous province and its application in prospectivity mapping. *Artificial Intelligence in Geosciences*, *2*, 60-75.

Zhu, L., Zhou, X., & Zhang, C. (2021). Rapid identification of high-quality marine shale gas reservoirs based on the oversampling method and random forest algorithm. *Artificial Intelligence in Geosciences*, *2*, 76-81.

Zou, H., & Hastie, T. (2005). Addendum: Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Series B Stat. Methodol.*, *67*(5), 768-768.

Los datos sísmicos fueron proporcionados por la Red Acelerográfica del Instituto de Ingeniería (RAII-UNAM), producto de las labores de instrumentación y procesamiento de la Unidad de Instrumentación Sísmica. Los datos son distribuidos a través del Sistema de Base de Datos Acelerográficos en web:
http://aplicaciones.iingen.unam.mx/AcelerogramasRSM/