

EVIDENCIAS EN PEDIATRÍA

Toma de decisiones clínicas basadas en las mejores pruebas científicas

www.evidenciasenpediatria.es

Fundamentos de medicina basada en la evidencia

Funcionamiento de los algoritmos de aprendizaje automático

Molina Arias M

Servicio de Gastroenterología Pediátrica. Hospital Universitario Infantil La Paz. Madrid. España.

Correspondencia: Manuel Molina Arias, mma1961@gmail.com

Palabras clave en español: algoritmo; aprendizaje automático; función de coste; métricas de desempeño; modelo.

Palabras clave en inglés: algorithm; machine learning; cost function; performance metrics; model.

Fecha de recepción: 31 de octubre de 2025 • **Fecha de aceptación:** 14 de noviembre de 2025

Fecha de publicación del artículo: 26 de noviembre de 2025

Evid Pediatr. 2025;21:50.

CÓMO CITAR ESTE ARTÍCULO

Molina Arias M. Funcionamiento de los algoritmos de aprendizaje automático. Evid Pediatr. 2025;21:50.

Para recibir Evidencias en Pediatría en su correo electrónico debe darse de alta en nuestro boletín de novedades en
<http://www.evidenciasenpediatria.es>

Este artículo está disponible en: <http://www.evidenciasenpediatria.es/EnlaceArticulo?ref=2025;21:50>.

©2005-25 • ISSN: 1885-7388

Este es un artículo Open Access bajo la licencia

CC BY-NC-ND (Reconocimiento-No comercial-Sin obras derivadas): <https://creativecommons.org/licenses/by-nc-nd/4.0/deed.es>

Funcionamiento de los algoritmos de aprendizaje automático

Molina Arias M

Servicio de Gastroenterología Pediátrica. Hospital Universitario Infantil La Paz. Madrid. España.

Correspondencia: Manuel Molina Arias, mma1961@gmail.com

INTRODUCCIÓN

Como ya mencionamos en artículos previos de esta sección de Fundamentos de Medicina Basada en la Evidencia^{1,2}, se ha producido un crecimiento exponencial del uso de la inteligencia artificial (IA) y de las técnicas de aprendizaje automático (ML, por sus siglas en inglés) en medicina³. El ML permite a los ordenadores aprender patrones a partir de datos sin necesidad de ser programados explícitamente para cada tarea. En el ámbito de la Pediatría, el potencial del ML es enorme: desde mejorar la precisión diagnóstica en imágenes médicas hasta optimizar tratamientos y predecir riesgos clínicos, todo ello pudiendo reducir la carga de trabajo de los profesionales sanitarios⁴.

En este artículo revisaremos los fundamentos de los algoritmos de ML, diferenciando entre algoritmos y modelos, y explicando sus componentes esenciales. Se describe también el proceso de división de datos, la validación cruzada y el ajuste de modelos. Esta visión general proporciona herramientas para interpretar con mayor rigor los estudios que aplican modelos de IA en la práctica clínica.

¿QUÉ ES UN ALGORITMO Y EN QUÉ SE DIFERENCIA DE UN MODELO?

Un algoritmo es una secuencia finita y ordenada de pasos o instrucciones que se siguen para resolver un problema o realizar una tarea.

Este procedimiento general, que no es exclusivo del ML, permite generar un modelo cuando se enfrenta a un conjunto de datos. De este manera, el modelo será la instancia entrenada del algoritmo o, dicho de otra forma, el resultado final del entrenamiento, es decir, la representación aprendida que puede de realizar predicciones sobre nuevos casos⁵.

Para ilustrarlo, imaginemos un algoritmo que aprende a diagnosticar neumonía a partir de radiografías de tórax. El algoritmo en sí podría ser, por ejemplo, una red neuronal convolucional; tras entrenarlo con miles de radiografías etiquetadas (con el diagnóstico conocido), obtenemos un modelo entrenado. Ese modelo ya no necesita los datos originales para funcionar: recibe una radiografía nueva y estima si hay neumonía

o no basándose en los patrones que “aprendió” durante el entrenamiento.

COMPONENTES DE UN ALGORITMO DE APRENDIZAJE AUTOMÁTICO

De manera general, podemos entender un algoritmo como una función parametrizada cuyo objetivo es realizar predicciones sobre una determinada variable. Para conseguirlo, durante el entrenamiento aprende de los datos los valores óptimos que deben tener sus parámetros para minimizar el error de predicción (la diferencia entre el valor predicho y el real, que es conocido en los casos de aprendizaje supervisado).

El proceso de entrenamiento se basa en unos elementos fundamentales: la función de coste, la retropropagación-optimización, los parámetros e hiperparámetros y la métrica de evaluación o desempeño. Veamos con detalle cada uno de ellos.

1. Función de coste. Se encarga de evaluar el rendimiento del modelo comparando las predicciones con los valores reales en un problema de ML. Su objetivo es minimizar el error de predicción, con lo que sirve de guía al algoritmo de optimización para ajustar los parámetros del modelo y mejorar su rendimiento.

Existen numerosas funciones de coste según se trate de aprendizaje supervisado (regresión o clasificación) o no supervisado. Entre las más comunes están el error cuadrático medio para regresión, las funciones de entropía cruzada para clasificación, y las distancias euclidianas y del coseno para técnicas de agrupamiento no supervisado y de aprendizaje profundo⁶.

2. Retropropagación-optimización. Una vez evaluado el error de predicción, el algoritmo ajusta los parámetros utilizando técnicas de optimización. La retropropagación, algoritmo especialmente relevante en redes neuronales, calcula el gradiente de la función de coste respecto a cada parámetro; esto es, la magnitud y dirección del error de la que es responsable cada parámetro. Seguidamente, es el optimizador el que emplea esta información, utilizando métodos de gradiente descendente como Adam o RMSProp, para actualizar el valor de los parámetros sumándoles el producto del error por un

hiperparámetro denominado tasa de aprendizaje. Esto se repite en cada ciclo del algoritmo, disminuyendo progresivamente el error y mejorando su rendimiento.

3. Parámetros e hiperparámetros. Los parámetros son los valores internos que el modelo aprende automáticamente a partir de los datos durante el entrenamiento, como los pesos en una red neuronal. Por otro lado, los hiperparámetros son configuraciones del algoritmo que el usuario debe definir antes del entrenamiento, como la tasa de aprendizaje, el número de capas o el tamaño del lote en una red neuronal. La correcta selección y ajuste de estos hiperparámetros puede tener un impacto significativo en la calidad y precisión de las predicciones.

Para simplificar, podemos decir que parámetro es todo aquel cuyo valor se aprende durante el entrenamiento, mientras que hiperparámetro es aquel cuyo valor debe ser decidido por parte del investigador antes del entrenamiento.

4. Métricas de evaluación. Una vez entrenado el modelo, se utilizan diversas métricas para valorar su rendimiento, como el valor predictivo positivo (*precision* en inglés), la sensibilidad (*recall* en inglés), y el F1-score (la media armónica entre los dos anteriores) en clasificación, o el error absoluto medio o el error cuadrático medio en regresión. Estas métricas permiten comparar diferentes modelos y seleccionar el más adecuado para el problema clínico concreto⁷.

Es importante no confundir la métrica del modelo, utilizada para evaluar el rendimiento con datos de validación o prueba, con la función de coste, que se utiliza durante el entrenamiento para evaluar el error de predicción de cada ciclo del

algoritmo. Lógicamente, existen funciones que pueden servir para ambos propósitos.

Una vez descritos los componentes de un algoritmo, podemos comprender mejor el proceso de entrenamiento, tal como se muestra en la **Figura 1**.

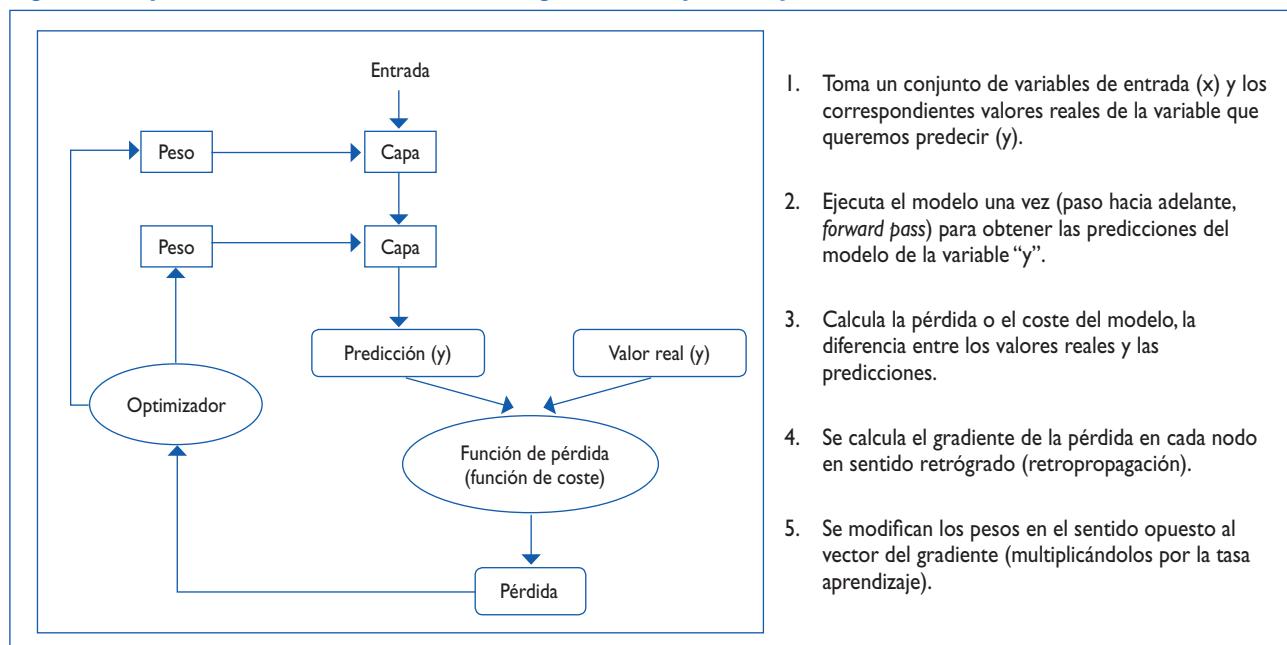
En primer lugar, se inicializan los valores de los parámetros, habitualmente de forma aleatoria. El algoritmo ejecuta los datos con esos valores de los parámetros y hace una predicción, que es evaluada por la función de coste. Como es lógico, la predicción y el valor real (conocido) serán diferentes y habrá un error de predicción. Este se transmite por el algoritmo de retropropagación, que determina la contribución al error de cada uno de los parámetros, permitiendo así al optimizador actualizar los parámetros a un nuevo valor. Una vez actualizados, el ciclo se repite de nuevo con los mismos datos, evaluando nuevamente el error y ajustando los parámetros, de forma que el error será cada vez menor tras cada ciclo. El entrenamiento terminará tras un número de ciclos establecido previamente (es otro hiperparámetro) o al alcanzar una condición de parada establecida, como una magnitud de error inferior a un umbral elegido.

EL EQUILIBRIO ENTRE SESGO Y VARIANZA

Este proceso de entrenamiento debe tener en cuenta un delicado equilibrio entre el sesgo y la varianza del modelo resultante.

El **sesgo** hace referencia a la tendencia del modelo a simplificar en exceso el problema, lo que puede provocar que no

Figura 1. Esquema de funcionamiento de un algoritmo de aprendizaje automático



capture patrones importantes de los datos (subajuste). Por el contrario, la **varianza** describe la sensibilidad del modelo ante pequeñas fluctuaciones en los datos de entrenamiento, lo que puede llevar a que el modelo se adapte demasiado a esos datos y pierda capacidad de generalización (sobreajuste). Lograr el equilibrio adecuado entre ambos es fundamental para obtener un modelo robusto y preciso.

Lo ideal será tener un modelo con bajo sesgo y baja varianza, pero no siempre es posible conseguir ambas cosas. A medida que aumenta la complejidad del modelo, el sesgo disminuye, pero a costa de un sobreajuste a los datos de entrenamiento, aumentando la varianza, lo que significa que disminuye también la capacidad de hacer predicciones cuando se enfrente con datos nuevos, que es el objetivo final por el que se elaboran los modelos⁸.

Para controlar este delicado equilibrio, se recurre a la división de los datos en tres conjuntos diferentes, como veremos a continuación.

DIVISIÓN DE LOS DATOS

Para conseguir el objetivo de que el modelo sea capaz de generalizar sus predicciones a datos nuevos no vistos durante el entrenamiento, es esencial subdividir los datos disponibles en tres subconjuntos: entrenamiento, validación y prueba⁹.

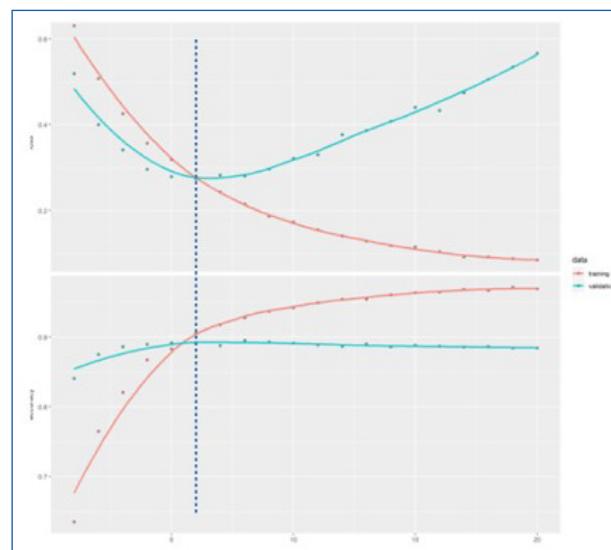
El conjunto de **entrenamiento** se utiliza para ajustar los parámetros del modelo, permitiendo que aprenda a partir de los datos, según el funcionamiento iterativo del algoritmo que ya hemos descrito.

El conjunto de **validación** sirve para evaluar el rendimiento durante el proceso de entrenamiento, de forma paralela a como se hace con los datos de entrenamiento. Así, tenemos dos entradas al algoritmo (entrenamiento y validación) y sus correspondientes salidas, que son evaluadas por la función de coste y la métrica de desempeño, después de cada iteración del entrenamiento.

Cuando el algoritmo está aprendiendo de los datos, la función de coste disminuirá con los dos conjuntos de datos, mientras que la de desempeño mejorará también en ambos. En el momento en que el algoritmo comience a sobreajustar los datos, veremos que con los datos de validación el error de la función de coste deja de disminuir o, incluso, aumenta, mientras que al desempeño le ocurre lo contrario: deja de mejorar o, incluso, disminuye (**Figura 2**). El modelo comienza a degradarse.

En este momento, o un poco antes, detendremos el entrenamiento. Así conseguiremos un modelo que quizás no haga un ajuste tan bueno de los datos de entrenamiento, pero que tendrá una mejor capacidad de generalizarse y de realizar

Figura 2. Representación gráfica del proceso de entrenamiento de una red neuronal artificial



Se muestran en rojo las curvas de entrenamiento y en azul las de validación. En la parte superior se muestran los errores de precisión, mientras que la inferior muestra la métrica de desempeño. Podemos ver que, con los datos de entrenamiento el desempeño mejora casi hasta el máximo y el error disminuye casi hasta el mínimo a lo largo del proceso. Sin embargo, las curvas de validación muestran que el modelo se degrada después de 5-6 iteraciones: el desempeño no mejora más (de hecho, disminuye algo) y el error invierte la tendencia previa y comienza a aumentar. El modelo comienza a hacer un sobreajuste a los datos de entrenamiento

predicciones con datos nuevos. O sea, minimizaremos el sesgo tratando que no aumente la varianza del modelo.

Por último, el conjunto de **prueba** se reserva exclusivamente para la evaluación final, proporcionando una estimación fiable de la capacidad de generalización del modelo sobre datos nunca vistos. Esta estrategia de partición resulta crucial para garantizar que el modelo sea verdaderamente útil en situaciones reales, donde los datos futuros pueden diferir significativamente de los utilizados durante el entrenamiento.

VALIDACIÓN CRUZADA

Cuando el volumen de datos no es suficiente para la división descrita, puede dividirse en dos subconjuntos (entrenamiento y prueba), utilizando uno de ellos para hacer una **validación cruzada**, que es una técnica avanzada que permite evaluar la capacidad de generalización del modelo de manera más robusta⁹.

Consiste en dividir los datos de entrenamiento en un número k de subconjuntos. El modelo se entrena repetidamente k veces utilizando en cada ciclo uno de los bloques como conjunto de validación y el resto como conjunto de entrenamiento, de modo que cada subconjunto actúa como conjunto de validación una vez. El rendimiento final se obtiene promediando los resultados de las k iteraciones, lo que proporciona una

estimación más fiable y menos dependiente de una única partición de los datos (**Figura 3**).

Como ya hemos dicho, este procedimiento es especialmente útil cuando se dispone de un número limitado de datos, ya que permite aprovechar al máximo toda la información disponible sin sacrificar la objetividad de la evaluación. Además, ayuda a detectar posibles problemas de sobreajuste o subajuste y facilita la selección del modelo más adecuado antes de realizar la evaluación final con el conjunto de prueba¹⁰.

AJUSTE DE HIPERPARÁMETROS

El ajuste de hiperparámetros es otro paso importante en el proceso de construcción de modelos de ML⁹.

Para encontrar la combinación óptima de hiperparámetros existen diversas estrategias, como la **búsqueda en cuadrícula** (grid search), la **búsqueda aleatoria** (random search) o métodos más avanzados, como la **optimización bayesiana**.

La búsqueda en cuadrícula define una serie de valores posibles para los hiperparámetros del algoritmo y explora todas las combinaciones posibles con un conjunto de datos definido. Por otra parte, con la búsqueda aleatoria se seleccionan al azar combinaciones de hiperparámetros dentro de un rango especificado y se evalúa el rendimiento del modelo con cada configuración.

Con cualquiera de estos métodos, se realiza una validación cruzada para evaluar el rendimiento de cada combinación, seleccionando aquella que ofrezca los mejores resultados en

el conjunto de validación y evitando así el sobreajuste. Un ajuste adecuado de los hiperparámetros puede marcar la diferencia entre un modelo mediocre y uno capaz de generalizar correctamente a nuevos datos.

CONCLUSIONES

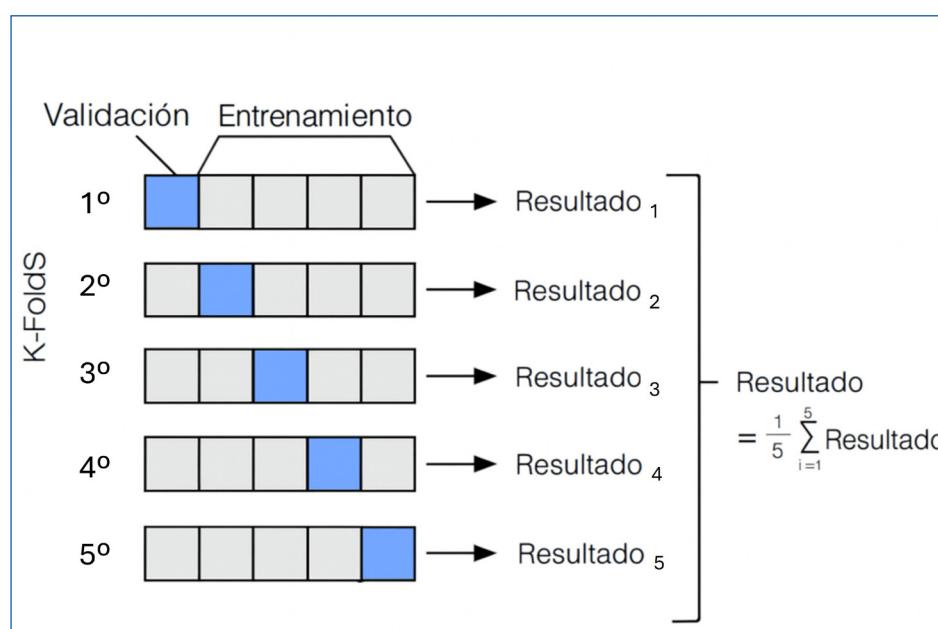
El ML representa una revolución silenciosa que ya está transformando diversos aspectos de la Pediatría, desde el diagnóstico por imagen hasta la monitorización domiciliaria y la investigación clínica, por lo que comprender el funcionamiento de los algoritmos de ML es fundamental para interpretar correctamente los estudios que los emplean en el contexto clínico.

La transparencia en los procesos de entrenamiento, validación y prueba, así como la selección adecuada de métricas son claves para garantizar modelos fiables y útiles. Un pediatra informado podrá interactuar mejor con estas herramientas, interpretar sus salidas con sentido crítico, detectar posibles sesgos y exigir validaciones adecuadas antes de tomar una decisión sobre su incorporación a la práctica clínica¹¹⁻¹³.

BIBLIOGRAFÍA

1. Molina Arias M. Inteligencia artificial en Pediatría: de la ciencia ficción a la realidad clínica. Evid Pediatr. 2025;21:11.
2. Molina Arias M. Redes neuronales artificiales: fundamentos y aplicaciones. Evid Pediatr. 2025;21:25.
3. An Q, Rahman S, Zhou J, Kang JJ. A comprehensive review on machine learning in healthcare industry: classification, restrictions, opportunities and challenges. Sensors (Basel). 2023;23:4178.

Figura 3. Esquema de validación cruzada con 5 bloques (k = 5).



4. Habehh H, Gohel S. Machine learning in healthcare. *Curr Genomics.* 2021;22:291-300.
5. Chollet F, Kalinowski WT, Allaire JJ. Deep learning with R. 2^a Ed. New York: Manning Publications Co; 2022.
6. Yu K, Beam AL, Kohane IS. Artificial intelligence in healthcare. *Nature Biomed Engin.* 2018;2:719-31.
7. Ghassemi M, Oakden-Rayner I, Beam AL. The false hope of current approaches to explainable artificial intelligence in health care. *Lancet (Digital Health).* 2021;3:E745-50.
8. Sendak M, D'Arcy J, Kashyap S, Gao M, Nichols M, Corey K, Ratliff W. A path for translation of machine learning products into healthcare delivery. *BMJ Health & Care Informatics.* 2020;27:e100109.
9. He J, Baxter SL, Xu J, Zhou X, Zhang K. The practical implementation of artificial intelligence technologies in medicine. *Nature Medicine.* 2019;25:30-3.
10. Demsar J, Zupan B. Hands-on training about overfitting. *PLoS Comput Biol.* 2021;17:e1008671.
11. Al-Zaiti SS, Alghwiri AA, Hu X, Clermont G, Peace A, Macfarlane P, et al. A clinician's guide to understanding and critically appraising machine learning studies: a checklist for Ruling Out Bias Using Standard Tools in Machine Learning (ROBUST-ML). *Eur Heart J Digit Health.* 2022;3:125-40.
12. Faes I, Liu X, Wagner SK, Fu DJ, Balaskas K, Sim DA, et al. A clinician's guide to artificial intelligence: how to critically appraise machine learning studies. *Trans Vis Sci Tech.* 2020;9:7.
13. Vinny PW, Garg R, Srivastava MVP, Lal V, Vishnu VY. Critical appraisal of a machine learning paper: a guide for the neurologist. *Ann Indian Acad Neurol.* 2021;24:481-9.