Fundamentos de machine learning

Óscar González Fernández

Diciembre 2020

Contenido

[Fundamentos de machine learning 1](#_Toc58759000)

[¿Qué es *machine learning*? 1](#_Toc58759001)

[Ejemplo 1](#_Toc58759002)

[Opción 1: sistema basado en reglas 2](#_Toc58759003)

[Opción 2: aprendizaje automático 2](#_Toc58759004)

[*Machine learning* en perspectiva 3](#_Toc58759005)

[Inteligencia artificial 5](#_Toc58759006)

[Conceptos de *machine learning* 6](#_Toc58759007)

[Caso de uso/Caso de negocio/Aplicación/Problema práctico 6](#_Toc58759008)

[Tarea (de aprendizaje) 7](#_Toc58759009)

[Técnica/Método 9](#_Toc58759010)

[Selección del modelo y sus hiperparámetros 12](#_Toc58759011)

# Fundamentos de machine learning

## ¿Qué es *machine learning*?

La invención del término *machine learning* (abreviado ML y en español *aprendizaje automático*) normalmente se ubica alrededor del año 1959 y se atribuye a [Arthur Samuel](https://en.wikipedia.org/wiki/Arthur_Samuel), uno de los científicos pioneros en la disciplina. Aunque el origen exacto es incierto, a día de hoy podríamos definirlo de esta manera:

El campo de estudio (arte y/o ciencia) que proporciona a los ordenadores la capacidad de aprender a partir de los datos, sin ser programados explícitamente para ello.

Es útil cuando **no se puede utilizar una fórmula** para describir la realidad, pero sí **se dispone de datos** para construir una aproximación empírica.

### Ejemplo

Supongamos que queremos desarrollar un sistema que, dada una imagen de un animal, pueda identificar si se trata de un gato o no.

Para conseguirlo, tenemos dos opciones principales.

### Opción 1: sistema basado en reglas

La primera opción sería la alternativa tradicional para resolver el problema: desarrollar un sistema basado en reglas.

Para ello, en primer lugar enumeramos las características propias de un gato como son:

* Ojos grandes, ovalados y ligeramente inclinados.
* Mentón prominente.
* Cabeza relativamente pequeña y en forma de cuña redondeada.
* Cola gruesa y uniforme con extremo redondeado y de longitud media.
* Pelo corto o de longitud media y denso. Suave y sedoso al tacto. Patrón a manchas o marmolado.
* Patas traseras de mayor longitud que las delanteras.
* …

Y, posteriormente, procedemos a programar un sistema basado en dichas reglas. Por ej:

* Si la cola del animal tiene un grosor entre 1 y 3 cm y su longitud está entre 10 y 40 cm, entonces es un gato.
* O, si las patas traseras son más largas que las delanteras y su pelaje muestra un patrón a manchas, entonces es un gato.
* …

Esta aproximación al problema tiene el inconveniente de que es muy laboriosa (requiere conocer muy bien las características distintivas de la especie y programar todas las reglas). Dado el gran trabajo que requiere conseguir una descripción bien completa de la especie, seguramente terminemos con una descripción parcial y nuestra solución puede no generalizar adecuadamente a gatos con características ligeramente diferentes a lo habitual.

### Opción 2: aprendizaje automático

Otra opción, es proporcionarle a un sistema de aprendizaje automático un número considerable de imágenes de gatos y otros animales y dejar que el sistema aprenda a distinguirlos. Es importante destacar que el sistema puede alcanzar una solución satisfactoria pero, en el proceso, no tiene por qué obtener necesariamente una representación explícita de las características utilizadas para distinguirlos.

La principal ventaja de este tipo de solución es que, si necesitamos que el sistema sea capaz de distinguir otros tipos de gatos, será suficiente con introducir nuevas imágenes y dejar que el sistema *aprenda* cómo distinguirlos.

Aunque no ocurre en todos los sistemas de machine learning, algunos tienen la desventaja de que cuando los utilizamos perdemos la interpretabilidad y explicabilidad del sistema:

* **Interpretabilidad**: es la capacidad que, como humanos, tenemos para entender la relación entre la entrada y la salida del sistema. Dicho de otra manera, es la capacidad para anticipar cuál va a ser la salida del sistema dado un cambio en la entrada o los parámetros que rigen el funcionamiento del sistema. En resumen, la interpretabilidad trata sobre comprender la mecánica del sistema desde fuera pero no requiere entender el porqué de su funcionamiento ni sus detalles internos.
* **Explicabilidad**: es la capacidad que tenemos para describir el funcionamiento del sistema con el objetivo de ser comprendido por otros humanos. Para esto, necesitamos conocer los detalles internos de funcionamiento del sistema y, basándonos en ellos, explicar qué está realmente sucediendo en su interior.

Habitualmente, ambos conceptos se utilizan de forma intercambiable y son importantes para responder a preguntas sobre la transparencia, auditabilidad o la prevención de posibles riesgos asociados a la utilización del sistema.

Ahora que hemos definido en qué consiste el aprendizaje automático podemos pensar en tecnologías que utilizamos en nuestro día a día que, por su funcionamiento, podrían estar utilizando aprendizaje automático.

Aquí algunos ejemplos:

* El buscador de Google.
* Los sistemas de reconomiento automático de matrículas de vehículos en carreteras o aparcamientos.
* La extracción automática de texto en documentos fotografiados con la cámara de nuestro móvil.
* El reconocimiento facial en las fotografías que subimos a Facebook.
* El recomendador de series o películas en Netflix.
* Los traductores de idiomas como [Google Translate](https://translate.google.com/) o [DeepL](https://www.deepl.com/en/translator).
* El trazado de rutas o las estimaciones del consumo calórico en [Strava](https://www.strava.com/).

## *Machine learning* en perspectiva

Como todos los ámbitos del conocimiento, el aprendizaje automático no es un campo aislado sino que está relacionado con otros campos del conocimiento. La siguiente figura muestra la relación entre los conceptos de *data science*, *artificial intelligence*, *machine learning*, *deep learning* y *big data*. Algunos autores prefieren utilizar otras clasificaciones pero en este material adoptaremos las siguientes definiciones:

* *Machine learning* es el subconjunto de *artificial intelligence* que se ocupa del aprendizaje a partir de ejemplos o experiencia. Su empleo se comenzó a popularizar alrededor de los años ochenta con éxitos destacables en la detección de correo electrónico no deseado (*spam*).
* *Deep learning* es, a su vez, un subconjunto de *machine learning* que alcanza grandes capacidades y flexibilidad aprendiendo a representar la realidad como una jerarquía de conceptos, donde cada concepto está definido en relación a otros conceptos más simples y donde se alcanzan conceptualizaciones cada vez más abstractas en base a conceptos más concretos. Hoy en día, esta línea está viviendo una gran explosión de uso desde que se popularizó alrededor de 2010 y está dominada por las redes neuronales artificiales de varias capas.
* *Artificial intelligence*, AI, o inteligencia artificial, es el campo que se ocupa de que las máquinas puedan realizar tareas que normalmente asociamos con la inteligencia humana. En la siguiente sección veremos los diferentes rasgos de la inteligencia humana y entenderemos por qué el aprendizaje representa solo una pequeña parte de ella. Es una disciplina que empezó a florecer en los años cincuenta y que, tradicionalmente, ha estado asociada al desarrollo de autómatas capaces de vencer a humanos en juegos tan famosos como las damas, el ajedrez o el go pero que, hoy en día, puede aplicarse a multitud de situaciones distintas.
* *Data science* o ciencia de datos es todo conocimiento científico aplicado al trabajo con datos.
* *Big data* es un concepto que empezó a popularizarse en los años 90 y se atribuye a [John Mashey](https://en.wikipedia.org/wiki/John_Mashey). Se define como el ámbito de conocimiento en la intersección de todas las anteriores disciplinas y las tecnologías necesarias para afrontar los nuevos retos que imponen las características de los “grandes” datos. Dichas características quedan condensadas en las dimensiones conocidas como *las 3 V’s del big data* y define por qué los datos pueden ser considerados “grandes”:
  + Volumen: disponemos de un **gran volumen** de datos.
  + Velocidad: se generan nuevos datos a una **gran velocidad**.
  + Variedad: los datos proceden de fuentes muy diversas y tienen una **gran variedad**.
* Hay quien prefiere añadir más V’s a la lista, como veracidad y valor. Sin embargo, dichas dimensiones no son definitorias del concepto de *big data* sino que son características imprescindibles también cuando se trabaja con datos que no cumplen ninguna de las primeras 3 V’s (algo que, por contraposición, podríamos denominar *small data*) y, por ello, no dan lugar a conocimiento ni tecnologías específicas del *big data*.

Es importante notar que, en ocasiones, las técnicas de *machine learning* requieren adaptarse para hacer frente a los retos del *big data* y el tipo de adaptaciones pueden ser muy diversas. Aquí algunos ejemplos:

* Existen algoritmos concebidos para ser ejecutados de manera distribuida (utilizando varias máquinas) para salvar la limitación de que los datos, por su gran volumen, no puedan almacenarse en una sola máquina o su procesamiento no pueda completarse en un tiempo razonable en una sola máquina.
* La mayoría de las técnicas de *machine learning* están concebidas para procesar de una sola vez la totalidad de los datos, lo que se conoce como procesamiento por lotes (*batch*). Sin embargo, para hacer frente a grandes volúmenes de datos o datos que llegan a gran velocidad, algunas técnicas se han adaptado para trabajar de manera incremental sobre cada uno de los nuevos datos que recibe (procesamiento *online*) o sobre pequeños lotes de datos (procesamiento en *mini-batches*).
* Se han mejorado las implementaciones de muchos algoritmos para proporcionar resultados exactos de manera más rápida o proporcionar resultados aproximados en un tiempo razonable y así poder enfrentarse a grandes volúmenes de datos.
* … y muchas otras más que se nos puedan ocurrir o que están por descubrir.

### Inteligencia artificial

Como hemos dicho anteriormente, se engloban en el concepto de inteligencia artificial todos los sistemas que son capaces de realizar tareas que normalmente asociamos con la inteligencia humana. Pero… ¿cuáles son dichas tareas? A continuación se incluye una lista no exhaustiva de las más relevantes:

* Solución de problemas:
  + Búsqueda clásica: dirigirse hacia un objetivo cuando lo único que importan son las decisiones propias.
  + Búsqueda frente a adversarios: dirigirse hacia un objetivo teniendo en cuenta las decisiones de otros (incluyendo el hecho de que éstos puedan actuar en nuestra contra).
  + Optimización: mejora de la solución a un problema.
* Gestión del conocimiento:
  + Obtención y extracción de conocimiento de datos númericos, texto, imágenes, audio o vídeo.
  + Representación del conocimiento en forma de entidades con sus interrelaciones.
  + Fusión de información aparentemente no relacionada o contradictoria.
* Razonamiento:
  + Lógico: razonamiento automático a partir de información veraz.
  + Difuso: razonamiento probabilístico en base a información de veracidad cuestionable.
* Interacción:
  + Social: habilidad de comunicarse con otros individuos por medio del lenguaje, la expresión de sentimientos, el comportamiento…
  + Con el mundo: percepción del mundo, movimiento, manipulación de objetos… Esta dimensión da lugar a lo que, desde un punto de vista filosófico se denomina *embodiment* y hace referencia a cómo el sistema perceptual y motor de un organismo condiciona los aspectos cognitivos del mismo y, por tanto, su inteligencia. De esta dimensión se ocupa, en gran medida, la robótica. Un buen ejemplo de cómo el sistema motor afecta al rendimiento puede verse en estos vídeos: [parte 1](https://www.youtube.com/watch?v=tIIJME8-au8), [parte 2](https://www.youtube.com/watch?v=lv6op2HHIuM).
* Aprendizaje: capacidad de mejorar el rendimiento en la realización de una tarea a medida que se acumula experiencia pasada.

Por tanto, denominamos inteligencia artificial a un programa de computación que se ha diseñado para realizar alguna de las operaciones que se consideran propias de la inteligencia humana como las que figuran en la lista anterior. Nótese que, aunque habitualmente atribuimos inteligencia a tareas que nos parecen complicadas (como jugar al ajedrez), éstas son relativamente sencillas de realizar para una inteligencia artificial que se ha especializado en esa tarea (*narrow AI*). Sin embargo, hay tareas mucho más sencillas (por ej. preparar un café, conocida como [Test de Wozniak](https://www.pcworld.com/article/134826/article.html)) que requieren de la combinación de más dimensiones de la inteligencia.

Se denomina inteligencia artificial general (*general AI*) a aquella que es capaz de realizar cualquiera de las tareas atribuibles a la inteligencia con un desempeño equivalente al de un ser humano. En el momento que las capacidades de esa inteligencia artificial sobrepasen las de un ser humano diremos que se ha alcanzado la superinteligencia. Nótese que, a día de hoy, ya existen máquinas cuyo desempeño es superior al de un ser humano al realizar algunas de estas tareas. Sin embargo, al tratarse de tareas concretas, esto no significa que posean superinteligencia.

## Conceptos de *machine learning*

En esta sección se hará un recorrido por los conceptos y terminología genérica más habituales utilizados por los científicos de datos en la disciplina del aprendizaje automático. Es importante entender y dominar estos conceptos ya que nos dotarán del lenguaje necesario para comprender el estado actual de la disciplina y continuar ampliando conocimientos en el futuro. Serán también una herramienta clave para comunicarnos con otros entendidos en la materia.

Nota aclaratoria: para cada uno de los conceptos que se describen a continuación se ha intentado incluir los sinónimos más habituales. En la vida real, sin embargo, pueden utilizarse otras variaciones y, en ocasiones, utilizarse la terminología de manera más laxa, imprecisa y llegando incluso a intercambiar unos términos por otros. Afortunadamente, en la mayoría de ocasiones, el contexto en el que se utiliza cada término es suficiente para aclarar su significado. Por esta razón, los contenidos en lo que resta de documento deben de ser entendidos como una formalización e interpretados de manera flexible al transponerlos al lenguaje cotidiano.

### Caso de uso/Caso de negocio/Aplicación/Problema práctico

El aprendizaje automático puede utilizarse para dar respuesta a muchos problemas que surgen en el mundo real, habitualmente como parte de la actividad cotidiana de las unidades de negocio en una empresa. Sin embargo, los problemas reales no suelen estar formulados de manera que se pueda aplicar ML directamente y es responsabilidad del científico de datos comprender en profundidad el problema, determinar los datos necesarios para llevar a cabo el proyecto y evaluar de qué manera el problema real se puede enmarcar en una o varias [**tareas de aprendizaje**](#Tarea).

Para medir el rendimiento de la solución aportada, normalmente, las unidades de negocio prefieren utilizar un [**KPI (*Key Performance Indicator*)**, **indicador o medida de desempeño**](https://en.wikipedia.org/wiki/Performance_indicator) que tenga sentido desde el punto de vista empresarial. Es también responsabilidad del científico de datos comprender esos indicadores para alinear debidamente los resultados de la solución de ML.

#### Ejemplo

Imaginemos que Netflix ha identificado recientemente un aumento en el número de bajas de usuarios y pretende identificar a aquellos usuarios que son más susceptibles de darse de baja en las próximas semanas. Este es uno de los casos de uso más habituales y se conoce con el nombre de [*churn prediction / attrition prediction*](https://en.wikipedia.org/wiki/Customer_attrition). El objetivo de negocio es contactar a dichos clientes para entender las razones por las que abandonan la plataforma y tomar posibles acciones al respecto. La solución aportada deberá, por tanto, identificar adecuadamente a esos clientes y contribuir a mejorar el KPI definido por el área de negocio. En este caso, la métrica que se utilizará para medir el éxito de la iniciativa será la tasa de cancelación de clientes ([*churn rate / attrition rate*](https://en.wikipedia.org/wiki/Churn_rate)).

### Tarea (de aprendizaje)

A pesar de tratarse de una disciplina de carácter incuestionablemente práctico, el conocimiento científico-técnico en ML no se desarrolla pensando en problemas reales sino en tareas de aprendizaje abstractas cuyo alcance está bien delimitado. De esta manera, el conocimiento desarrollado puede aplicarse a multitud de problemas reales distintos. Como decíamos anteriormente, el científico de datos será quien deba determinar qué tarea o tareas de aprendizaje son de aplicación a su caso de uso.

Al tratarse de tareas de aprendizaje, todas hacen uso de ejemplos de datos, pero en función de qué datos se utilizan y cómo se utilizan pueden distinguirse varios tipos.

#### Tipos de aprendizaje

Aprendizaje supervisado (*Supervised learning*)

En el aprendizaje supervisado se hace uso de ejemplos de datos de entrada y ejemplos de datos de salida. De forma equivalente se dice que se trabaja con datos etiquetados porque se utilizan las “etiquetas” o datos de salida asociados a cada uno de los datos de entrada. El objetivo es aprender sobre la distribución que los relaciona.

Aprendizaje no supervisado (*Unsupervised learning*)

En el aprendizaje no supervisado solamente se utilizan ejemplos de datos de entrada (no etiquetados). El objetivo es aprender sobre la distribución de dichos datos.

Aprendizaje semi-supervisado (*Semi-supervised learning*)

El aprendizaje semi-supervisado combina datos etiquetados con datos no etiquetados. Normalmente, dada la dificultad para obtener datos etiquetados en algunos contextos (se suele necesitar un experto humano para etiquetar los datos), se utiliza una pequeña cantidad de datos etiquetados para mejorar el rendimiento del aprendizaje no supervisado.

Aprendizaje con refuerzo (*Reinforcement learning*)

En el aprendizaje con refuerzo no se utilizan ejemplos etiquetados o no etiquetados. En su lugar, el sistema explora el entorno y obtiene recompensas o penalizaciones en función de las acciones tomadas. Para conseguir que el sistema aprenda adecuadamente es muy importante fijar correctamente el sistema de recompensas/penalizaciones. El objetivo es que el sistema aprenda la política de acciones que debe tomar en cualquier situación para maximizar la recompensa total acumulada. Es el paradigma de aprendizaje que mejor modeliza problemas como la [conducción autónoma](https://medium.com/@SmartLabAI/controlling-self-driving-robots-with-deep-reinforcement-learning-3b60bb0a9ea7) o [aprender a caminar](https://deepsense.ai/learning-to-run-an-example-of-reinforcement-learning/).

#### Tipos de tareas

Clasificación

Las tareas de clasificación tienen como salida una variable discreta. Si la variable de salida es binaria, la tarea se denomina [clasificación binaria](https://en.wikipedia.org/wiki/Binary_classification). Este es un tipo de clasificación muy habitual en diagnóstico médico donde se debe determinar si un paciente tiene cierta enfermedad o no a partir de los datos recogidos en una o varias pruebas diagnósticas.

En caso de que la variable de salida pueda tomar más de dos valores distintos se denomina [clasificación multiclase](https://en.wikipedia.org/wiki/Multiclass_classification). Un ejemplo típico de clasificación multiclase es el reconocimiento de dígitos manuscritos donde, a partir de la imagen de un dígito, se debe determinar a cuál de los 10 posibles dígitos se refiere la imagen.

Cuando la salida puede tomar varios valores distintos simultáneamente porque no son mutuamente excluyentes, la clasificación se denomina [clasificación multi-etiqueta](https://en.wikipedia.org/wiki/Multi-label_classification). Una muestra de clasificación multi-etiqueta sería la clasificación de textos según los temas que se tratan donde, por ejemplo, se puede hablar de política y medio ambiente simultáneamente.

Regresión

Las tareas de clasificación tienen como salida una variable continua. Un ejemplo puede ser predecir la temperatura que va a hacer mañana, o la nota de un examen.

Existe un caso particular de regresión a mitad de camino entre la regresión y la clasificación que se suele denominar [regresión ordinal](https://en.wikipedia.org/wiki/Ordinal_regression) y es aquel donde el objetivo es predecir una variable ordinal. Aparece con frecuencia en las ciencias sociales al intentar determinar niveles de preferencia dentro de una escala (por ej. 1-5 donde 5 denota el más preferido).

Clustering

Las tareas de *clustering* consisten en agrupar una serie de objetos de tal forma que los objetos en el mismo grupo (denominado *cluster*) sean más similares entre sí que con objetos de distintos grupos. Un ejemplo típico es el de una compañía que quiere segmentar su base de clientes en función del nivel de vinculación que mantienen con la empresa, donde la vinculación viene determinada por el uso que hace de distintos productos.

Reducción de dimensionalidad

La reducción de dimensionalidad consiste en transformar los datos de un espacio de alta dimensionalidad a un espacio de dimensionalidad más baja. Habitualmente se busca que el resultado de baja dimensionalidad preserve las características más significativas de los datos originales. Un ejemplo típico es utilizarla para facilitar la visualización de conjunto de datos de alta dimensionalidad obteniendo una representación en 2 ó 3 dimensiones.

Detección de anomalías

La detección de anomalías consiste en la identificación de valores inusuales o atípicos (*outliers*) en un conjunto de datos. Se considera que un dato es inusual cuando difiere considerablemente de la mayoría. Un ejemplo típico es la detección de fraude en operaciones con tarjeta de crédito. Esta tarea puede también entenderse como un caso particular de clasificación denominada [clasificación unaria](https://en.wikipedia.org/wiki/One-class_classification).

Ranking

La tarea de *ranking* (o *learning to rank*) consiste en aprender a ordenar objetos de tal forma que guarden la mejor concordancia posible con una puntuación numérica o una valoración binaria (por ej. si el objeto es relevante o no). Un caso típico son los sistemas de búsqueda y recuperación de información como, por ejemplo, un buscador web. En estos sistemas, tras una consulta por parte del usuario, se presenta una lista de documentos ordenada según la relevancia que se estima que tienen para la consulta realizada.

#### Ejemplo

Retomemos el ejemplo de Netflix, en el que queríamos identificar a aquellos usuarios que son más susceptibles de darse de baja en las próximas semanas. Desde un punto de vista de ML podríamos enfocarlo como cualquiera de las siguientes tareas:

* Regresión: utilicemos la serie histórica de contenidos visionados semana a semana y predigamos el número de visionados que cada usuario hará en las próximas semanas. Si la intensidad de visionado predicha está por debajo de un cierto umbral para un determinado usuario consideraremos que dicho usuario está en riesgo de darse de baja pues ya no muestra interés en la plataforma.
* Clasificación: recojamos los patrones de visionado pasados de una muestra significativa de los usuarios. Dicha muestra tiene que contener tanto ejemplos de usuarios que ya se dieron de baja como usuarios activos. Basándonos en esta información histórica predigamos si el usuario se dará de baja o no.
* Detección de anomalías: identifiquemos a aquellos usuarios que, por sus patrones de uso, se desvían del comportamiento normal. Es probable que dentro de estos localicemos a clientes insatisfechos.

No existe una receta clara para seleccionar el tipo de tarea o tareas adecuadas. La elección concreta dependerá de nuestra experiencia, nuestra comprensión del problema real y del tipo, cantidad y calidad de los datos de los que dispongamos, entre otros.

### Técnica/Método

Las técnicas o métodos de *machine learning* son las herramientas utilizadas para resolver cualquiera de las anteriores tareas. La lista de ejemplos es inmensa y se amplía constantemente. Seguramente, el lector ya habrá oído hablar de muchos de ellos como son:

* Regresión lineal: para la tarea de regresión supervisada.
* Regresión logística: clasificación binaria supervisada.
* *Naive Bayes*: clasificación multiclase supervisada.
* *K-means*: clustering no supervisado.
* Árboles: regresión y clasificación supervisada.
* Máquinas de vectores de soporte (*Support Vector Machines*, SVMs): regresión y clasificación supervisada.
* *K-nearest neighbors* (KNN): clasificación/regresión/detección de anomalías supervisada.
* Análisis de componentes principales (*Principal Component Analysis*, PCA): reducción de dimensionalidad.
* Kernel PCA, redes neuronales, procesos gaussianos, LDA, ISOMAP, DBSCAN, etc.

Cada una de estas técnicas propone un **modelo matemático** distinto para describir la realidad de nuestros datos. Por ej.: “la variable de salida puede explicarse razonablemente bien como una combinación lineal de las variables de entrada” o “los datos puede ser separados razonablemente bien mediante un plano”.

Veamos también, con un ejemplo sencillo, de qué estamos hablando en términos matemáticos. Tomemos una regresión lineal univariante, donde se intenta explicar como una función lineal de . Es decir:

donde denota el valor estimado de obtenido por el algoritmo y denota la variable de entrada.

#### Entrenamiento/ajuste de parámetros

El objetivo del proceso de aprendizaje es obtener el valor óptimo para los **parámetros del modelo**, en este caso, los coeficientes y . Esto se consigue durante el proceso de **entrenamiento**.

Nota: Los métodos cuyo funcionamiento queda resumido en un conjunto de parámetros se denominan **métodos paramétricos**. Existen también **métodos no paramétricos** que no hacen asunciones fuertes sobre la forma que tendrá la relación entre la entrada y la salida y son, por tanto, libres de aprender cualquier tipo de relación cuando son alimentados con datos. Sin embargo, tienen la desventaja de que el resultado del aprendizaje no queda resumido en un conjunto reducido de parámetros y, en cambio, el método tiene que memorizar la totalidad de los datos de entrenamiento. Un ejemplo de método no paramétrico es *K-nearest neighbors*.

Durante el proceso de entrenamiento lo que se hace es optimizar una **función de coste**. En los algoritmos de regresión lineal lo habitual es minimizar el error cuadrático medio (*mean squared error*, MSE). En este caso, eso significa que durante la fase de entrenamiento lo que haremos será localizar los valores de y que minimizan la siguiente función de coste:

donde y denotan el valor de las variables de entrada y salida para el -ésimo ejemplo, respectivamente. denota el número total de ejemplos utilizados para el entrenamiento.

En la práctica dicho proceso de entrenamiento se lleva a cabo en un computador y, por consiguiente, lo que nosotros utilizamos no es un método de ML sino una **implementación software** de dicho método. Es en el momento de implementación cuando los métodos se convierten en secuencias de pasos que en el momento de la ejecución se llevarán a cabo de una manera ordenada. Por esta razón, las técnicas o métodos de ML se suelen denominar también **algoritmos** de ML, pero realmente hacen, a su vez, uso de otros algoritmos para las tareas más fundamentales.

Por ejemplo, para minimizar las funciones de coste, suele recurrirse a **algoritmos de optimización** como descenso por gradiente o, más comunmente, descenso por gradiente estocástico (*stochastic gradient descent*, SGD) o alguna de sus variantes. El repositorio [pytorch-optimizer](https://github.com/jettify/pytorch-optimizer) contiene una buena colección de algoritmos de optimización para utilizar con la biblioteca [pytorch](https://pytorch.org/). La decisión de utilizar un algoritmo de optimización u otro dependerá, por un lado, de la complejidad de la función de coste y por otro del volumen de datos a utilizar.

Este tipo de detalles son propios de la implementación y, en ocasiones, dan lugar a **parámetros de la implementación** que no juegan ningún papel en la definición del modelo matemático e incluso, en ocasiones, puede que no afecten ni siquiera al resultado del entrenamiento. Veámoslo con el ejemplo de la [implementación del modelo para regresión logística en sklearn](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html#sklearn.linear_model.LogisticRegression):

sklearn.linear\_model.LogisticRegression(penalty='l2', \*, dual=False,   
 tol=0.0001, C=1.0, fit\_intercept=True, intercept\_scaling=1,   
 class\_weight=None, random\_state=None, solver='lbfgs', max\_iter=100,   
 multi\_class='auto', verbose=0, warm\_start=False, n\_jobs=None, l1\_ratio=None)

En este caso, la mayoría de parámetros son parámetros de la implementación y están relacionados con el algoritmo de optimización utilizado en el entrenamiento. En concreto:

* solver: Selecciona el algoritmo de optimización a utilizar.
* tol: Indica un valor de tolerancia para detener la optimización.
* max\_iter: Indica el número máximo de iteraciones que se permitirán al algoritmo de optimización.
* dual: Selecciona la formulación del problema que deberá resolver el solver.
* n\_jobs: Indica el número de procesadores que se utilizará para la optimización.
* random\_state: Valor de semilla a utilizar para la generación de un punto de inicio aleatorio para los algoritmos de optimización. Permite garantizar la reproducibilidad de la ejecución.
* warm\_start: Indica si se deben utilizar los valores obtenidos en la anterior optimización como punto de inicio para siguientes optimizaciones.
* verbose: Indica si se debe mostrar el progreso del algoritmo de optimización.

Este tipo de parámetros no deben confundirse nunca con los parámetros del modelo, que no son proporcionados por el usuario sino que se obtendrán como resultado del proceso de entrenamiento, ni con los hiperparámetros del modelo, que veremos más adelante.

#### Predicción

Una vez se ha entrenado el modelo y se dispone de los valores de los parámetros, podemos utilizar dichos parámetros para obtener predicciones de la variable de salida para nuevos ejemplos de entrada. Es tan sencillo como evaluar la expresión matemática del modelo sobre los nuevos ejemplos :

Una vez obtenidas las predicciones podemos evaluar su calidad comparándolas con los valores reales de las mismas. Para hacer dicha comparación utilizaremos una o varias **métricas de bondad** (*goodness-of-fit metrics*).

Es habitual utilizar como métrica de bondad la misma que la función de coste del método pero no es necesario y, en ocasiones, no es incluso recomendable.

En este caso podríamos medir la calidad de las soluciones mediante el MSE, el MAE (*mean absolute error*), la correlación o cualquier otra métrica que consideremos interesante. Como norma general, deberíamos elegir aquellas que además de servirnos para establecer el correcto funcionamiento del método, guarden cierta conexión con los objetivos de negocio (o KPIs) que define el área de negocio para nuestro caso de uso. Hemos de tener siempre presente que estamos aplicando ML para resolver un problema real y, por tanto, nuestras evaluaciones deben de ser lo más aproximadas posibles a cómo se evaluará al modelo en el mundo real.

### Selección del modelo y sus hiperparámetros

En ocasiones, lo que venimos llamando modelo no describe un único modelo matemático sino que describe una familia parametrizada de modelos. Los parámetros que seleccionan cada uno de los miembros de dicha familia se denominan **hiperparámetros**.

Es importante notar que los hiperparámetros no se optimizan durante el proceso de entrenamiento y son, en cambio, fijados manualmente o seleccionados mediante un proceso de validación cruzada. Esta misma idea se puede extender para **seleccionar el modelo** más adecuado con candidatos de diferentes familias. Por ej. seleccionar entre un árbol de regresión y una regresión lineal.

#### Ejemplo

Volvamos por un momento al ejemplo anterior de la regresión lineal univariante. Ahora, pensemos que dicha regresión lineal pertenece a la familia de la regresión polinómica. Esto es, la familia de modelos donde el valor predicho se obtiene de la siguiente manera:

Dado este modelo, la regresión lineal anterior es un caso particular que se obtiene cuando . Pero, en esta ocasión, tenemos la libertad de seleccionar modelos de cualquier grado . Este parámetro es lo que se conoce como un **hiperparámetro del modelo**.