ACERCA DEL AUTOR

JESÚS BOBADILLA SANCHO

Doctor y Licenciado en Informática por la Universidad Politécnica de Madrid. Catedrático de Escuela en la Escuela Técnica Superior de Ingeniería de Sistemas Informáticos (ETSISI) de la Universidad Politécnica de Madrid (UPM). Ha impartido docencia en los campos de Arquitectura de Computadores, Sistemas Operativos, Telemática, Reconocimiento de Voz, Inteligencia Artificial y Programación Orientada a Objetos. Ha sido autor de diez libros publicados en el ámbito de la Informática. Su trayectoria de investigación incluye estancias en la Universidad de Berkeley y en la Universidad de Sheffield. Es autor de numerosas publicaciones técnicas en revistas internacionales de alto impacto, y que suman una gran cantidad de citas. Una de estas publicaciones se encuentra entre el 1% de las más relevantes a nivel mundial en el área de Informática (fuente: Web of Science). Su investigación principal se centra en el Filtrado Colaborativo que fundamenta los modernos Sistemas de Recomendación (Amazon, Netflix, Spotify, etc.): un campo importante en el área del Aprendizaje Automático.

1

INTRODUCCIÓN

En este capítulo se explican diversos conceptos importantes de machine learning (aprendizaje automático). Estos conceptos nos ayudarán a comprender los siguientes apartados, donde se verán modelos de regresión y clasificación, así como las técnicas de clustering (agrupamiento) y de reducción de dimensiones. Mientras que los siguientes capítulos incluirán algunos formalismos matemáticos y desarrollos en Python, aquí se explicarán los conceptos de la manera más simple posible. En resumen, este capítulo pretende ofrecer un marco que facilite la comprensión de los principales conceptos asociados a machine learning.

Machine learning es la ciencia que hace que los ordenadores "aprendan" a partir de los datos. En vez de programar, paso a paso, cada solución específica para cada necesidad planteada, tal y como se realiza en el enfoque de la programación convencional, el área de machine learning está dedicada al desarrollo de algoritmos genéricos que pueden extraer patrones de diferentes tipos de datos. De esta manera, un programa de machine learning destinado, por ejemplo, a clasificar números escritos a mano, no va a diferir sustancialmente de un programa destinado a la clasificación de las imágenes de señales de tráfico: ambos se basarán en la existencia de algún tipo de algoritmo de machine learning que clasifique datos etiquetados. En este punto se podría pensar que el proceso completo de machine learning es fácilmente automatizable, cuando realmente no es el caso: un ingeniero de datos (*data scientist*) debe llevar a cabo numerosas tareas específicas tales como la identificación de la fuente de datos, su limpieza, la eliminación de información que esté fuertemente correlacionada, la búsqueda de información sesgada, la realización de las normalizaciones necesarias, la identificación de los tipos de soluciones de machine learning cuya aplicación resulte apropiada, la elección del algoritmo más adecuado, el ajuste fino de los hiperparámetros del método elegido, el análisis de los resultados, la identificación de comportamientos incorrectos, la vuelta a procesos anteriores con el fin de cambiar lo que resulte necesario para mejorar los resultados, etc.

1.1 TIPOS DE MACHINE LEARNING

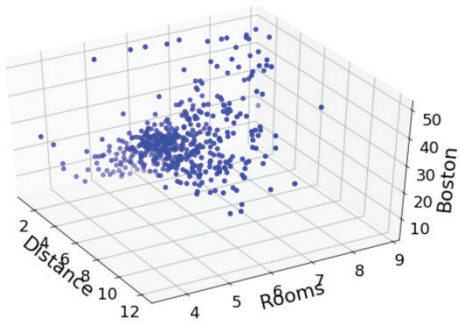
Con el objetivo de poder abordar cualquier tarea específica, el ingeniero de datos debe conocer algunos conceptos importantes de machine learning, así como las diferentes opciones existentes, las medidas de calidad más utilizadas, etc. Los conocimientos básicos incluyen la identificación de las tareas, empezando por la clasificación de los problemas de machine learning en alguno de los siguientes tipos:

- Aprendizaje supervisado
  - regresión
  - clasificación
- Aprendizaje no supervisado
  - clustering (agrupamiento)
  - reducción de dimensiones
- Aprendizaje semi-supervisado
- Aprendizaje por refuerzo

El aprendizaje supervisado en machine learning se aplica cuando cada dato, o conjunto de datos de entrada (muestra) tiene asociada una etiqueta. Pongamos un ejemplo: un conjunto de imágenes en las que cada una de ellas contiene algún tipo de metadato (habitualmente una etiqueta o conjunto de etiquetas): ((pict0001.bmp, "perro"), (pict0002.bmp, "pájaro"), (pct0003.bmp, "gato"). Partiendo de este conjunto de datos se pueden usar diferentes algoritmos de clasificación de machine learning con el objetivo de "entrenar" un modelo y poder, al acabar el entrenamiento, predecir la etiqueta correspondiente a una nueva imagen (no incluida en el conjunto de datos original); éste es un problema de clasificación. De igual manera, podemos hacer uso de un conjunto de datos que contenga muestras con valores numéricos asociados: por ejemplo, un conjunto de muestras de terremotos cuyos datos contienen la intensidad de la vibración previa tomada de sensores y cuyo objetivo es determinar la intensidad oficial del terremoto ([7.1, 6.3, ...], 5.4), ([3.2, 9.7, ...], 7.1). Este es un problema de regresión y su utilidad podría ser la de generar información acerca de la intensidad predicha por el modelo de regresión cuando se le aporta una nueva muestra (valores sísmicos recogidos en tiempo real).

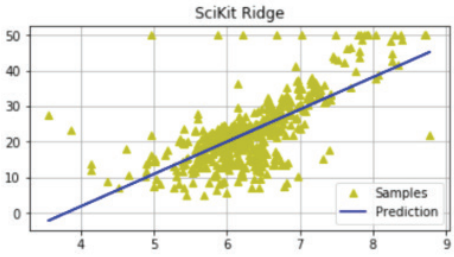
Los siguientes números escritos a mano pueden ayudar a entender las posibilidades de los algoritmos de clasificación. Como veremos más tarde, hemos conseguido reconocerlos todos con la excepción del 'cinco' situado en medio de los dos 'nueves'.

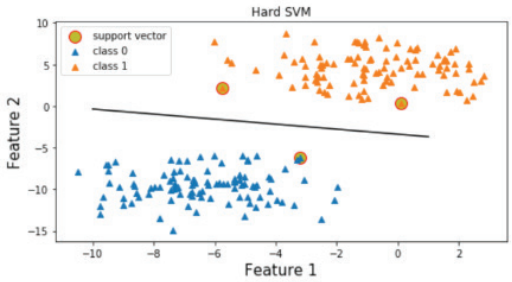
En el siguiente gráfico tridimensional podemos ver los datos correspondientes a un problema de regresión. El objetivo es predecir el precio de una propiedad inmobiliaria en Boston atendiendo a diferentes tipos de información. En el gráfico solo se muestra el "número de habitaciones" y la "distancia a la autopista" como tipos de información. El término utilizado en machine learning para los "tipos de datos de información" es "característica" ("*feature*"). Así, en el ejemplo, tanto "número de habitaciones" como "distancia a la autopista" son características. Los ejes horizontales *x* e *y* representan, cada uno, una característica. El eje vertical *z* muestra los valores objetivos para cada precio de venta de las propiedades inmobiliarias de Boston. En este caso, los valores objetivos no son etiquetas o nombres de categorías: son valores numéricos. Aquí afrontamos un problema de regresión, donde al suministrar un nuevo dato (habitaciones, distancia) lo que se obtiene es la predicción esperada: el precio de la propiedad inmobiliaria.



De los párrafos anteriores podemos entresacar un concepto importante: el modelo de machine learning. Este es un elemento clave, ya que la mayoría de los algoritmos de machine learning crean un modelo a partir de los datos. El modelo puede ser tan simple como la solución lineal que mejor ajuste las muestras de origen a los valores objetivo, o mucho más complejo, como la búsqueda de factores ocultos que representen la información más importante que esta contenida en los datos. La siguiente figura muestra una regresión lineal (gráfico de la izquierda) y una clasificación lineal (gráfico de la derecha). En el primer caso, los precios en Boston se obtienen usando un modelo de regresión simple que solamente usa la información más representativa de estos datos: el número de habitaciones. Para predecir el precio de una nueva propiedad inmobiliaria simplemente debemos conocer su número de habitaciones, y el modelo predecirá, linealmente, su valor. El gráfico de la derecha muestra dos clases generadas (clase 0 y clase 1), cada una de las cuales está definida por dos características (característica 1 y característica 2). El modelo de clasificación lineal ha resultado ser capaz de separar ambas clases. Para conocer la clase a la que

pertenece una nueva muestra usamos el modelo lineal: indica si la muestra está a un lado o a otro de la recta que separa ambas clases. Aquí, el algoritmo de clasificación ha deducido ("entrenado") un modelo a partir de los datos: la recta de color negro es el modelo aprendido.





La importancia del aprendizaje supervisado en machine learning está aumentando muy rápidamente debido a:

1. Las nuevas oportunidades brindadas por el *Internet de las Cosas* (Internet of Things o IoT), de donde se pueden obtener cantidades masivas de datos etiquetados de manera automática.

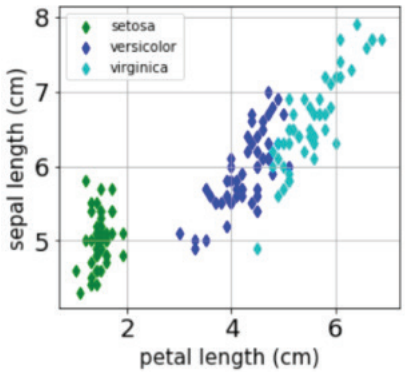
2. Las redes sociales, en cuyos servidores se almacena una enorme cantidad de interacciones y cuyo número de aplicaciones, como las destinadas al mercado digital, no para de crecer.

3. Los nuevos algoritmos, destinados a resolver diferentes tipos de aprendizaje supervisado, que hacen posible obtener resultados comerciales significativos: conducción automática, reconocimiento facial, sistemas de recomendación, etc.

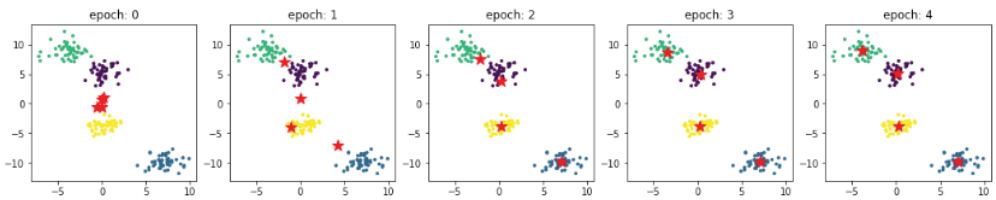
4. Las crecientes capacidades de procesamiento, particularmente las supercomputadoras paralelas y las unidades de procesamiento gráfico (GPUs).

5. La "democratización" del machine learning, por la que todos podemos trabajar con recursos altamente tecnológicos como Tensorflow o granjas de GPUs, así como con potentes APIs, entornos, IDEs, etc., tales como las tecnologías proporcionadas por *Scikit, Keras y Jupiter*.

El aprendizaje no supervisado utiliza información no etiquetada. La aplicación más conocida del aprendizaje no supervisado es la de clustering (agrupamiento). El objetivo de la técnica de clustering es agrupar muestras: p. ej.: para obtener los diferentes tipos de clientes en un servicio online, para agrupar productos en un comercio electrónico, para identificar comportamientos en la conducción, etc. El siguiente gráfico muestra un esquema típico de clustering. Contiene tres clusters (grupos o clases) correspondientes a tres diferentes tipos de lirios. Podemos observar que es fácil diferenciar el tipo "*setosa*" de los otros dos, mientras que se podrían presentar dificultades para precisar los grupos de los tipos *versicolor* y *virginica*.



Un modelo de clustering podría proporcionar varios hiperplanos lineales de separación (en el caso anterior: dos rectas), mientras que un modelo de clustering diferente podría proporcionar algunos elementos virtuales representativos: *centroides* cuya "área de influencia" determina a qué cluster pertenece cada una de las muestras. Hay más tipos de modelos de clustering, pero los que se han indicado pueden ayudar a entender el concepto de modelo y el hecho de que diferentes algoritmos de machine learning pueden estar basados en diferentes tipos de modelos. El siguiente gráfico muestra un ejemplo de la evolución de los centroides (estrellas rojas) de cara a ajustar los cuatro clusters existentes en los datos de ejemplo.



La reducción de dimensionalidad se usa habitualmente como una etapa de pre-procesamiento en algún otro tipo de labores de machine learning, principalmente en clasificación o regresión. Muchos escenarios reales aportan datos dispersos o datos que en su mayoría proporcionan muy poca información. Un ejemplo de datos dispersos es la información que se maneja en un sistema de recomendación: los usuarios solamente compran, hacen clic, consumen o votan una proporción muy pequeña de los productos, canciones, películas, etc., disponibles. Si colocamos esta información en forma de matriz (usuarios x ítems), la matriz contendrá una gran proporción de elementos sin información: normalmente más del 97% de los datos en sistemas de recomendación son así. Siendo capaces de comprimir los datos, tal y como se hace con las imágenes o con los ficheros de texto, los datos comprimidos contendrían casi toda la información, pero de una manera "condensada". Trabajar con esta información comprimida es mucho más eficiente y produce resultados más precisos. Lo que hacemos aquí es convertir datos multidimensionales (una dimensión para cada una de las características: para cada ítem), que son altamente dispersos, en información multidimensional mucho más concentrada y densa. La siguiente figura muestra una imagen fuente y varias versiones de la imagen obtenida a base de aplicar diferentes niveles de reducción de dimensionalidad.

El aprendizaje semisupervisado trata con conjuntos de datos en los que una porción de los datos está etiquetada y el resto no. Normalmente, la cantidad de muestras etiquetadas es mucho más pequeña que las no etiquetadas. La mayoría de los algoritmos de aprendizaje semisupervisado son una mezcla de métodos supervisados y no supervisados. El aprendizaje por refuerzo es un área innovadora y con un gran futuro, ya que está inspirada en mecanismos naturales. En este caso, el algoritmo de aprendizaje recibe información de un entorno real o simulado. Cuando el sistema realiza una acción es recompensado o penalizado, tal y como pasa con los seres vivos. Tales algoritmos de aprendizaje se denominan agentes y pueden aprender siguiendo los principios de la evolución natural. Los agentes aprenden estrategias, denominadas "políticas", que maximizan las recompensas y minimizan las penalizaciones. La mayoría de los sistemas de inteligencia artificial actuales, que están especializados en juegos, están basados en el enfoque de aprendizaje por refuerzo.

Los métodos de machine learning también pueden ser clasificados como:

- Basado en modelos o basados en memoria.
- Aprendizaje incremental o aprendizaje por lotes.
- Aprendizaje superficial (shallow learning) o aprendizaje profundo (deep learning).

Los algoritmos basados en memoria (basados en instancias) toman las muestras de datos como entrada y procesan directamente la predicción o la clasificación. Si se necesita una nueva predicción se procesa de nuevo, partiendo de las muestras de datos. Por el contrario, los algoritmos basados en modelos necesitan actualizar el modelo periódicamente, aunque el proceso de predicción es mucho más rápido que en el enfoque basado en memoria. La separación lineal que hemos visto previamente es un modelo simple: puede llevar algún tiempo calcular la pendiente y el punto de corte de la recta, pero la predicción del valor de y partiendo de una nueva muestra x es muy rápido. Si se aportan nuevas muestras de manera periódica, el modelo debe ser actualizado para poder calcular la nueva recta que se ajusta mediante clasificación o regresión.

Los algoritmos de aprendizaje por lotes (batch learning) siempre calculan el modelo desde el principio. Si disponemos de 2000 muestras, el proceso por lotes las usa todas para crear el modelo. Cuando se aporten 300 nuevas muestras al sistema, el proceso por lotes crea el modelo desde el principio, usando las 2300 muestras y así sucesivamente.

Los algoritmos de aprendizaje incremental no crean sucesivos modelos desde el principio: actualizan el modelo existente. En nuestro ejemplo lineal, los algoritmos de proceso por lotes obtendrían el modelo procesando, desde cero, las 2300 muestras existentes. Sin embargo, los algoritmos de aprendizaje incremental usarían las 300 nuevas muestras para cambiar los valores de la pendiente existente y el punto de corte (en muchas situaciones se usan más muestras de datos que las 300 nuevas). Los algoritmos incrementales presentan una importante ventaja: pueden ser usados como el núcleo de sistemas de machine learning escalables.

En el aprendizaje superficial (shallow learning), los parámetros (pendiente, punto de corte, etc.) se aprenden directamente de las características de las muestras de datos. En el aprendizaje profundo (deep learning) siempre existe una arquitectura con más de un nivel (capa). En el segundo nivel (y sucesivos), los parámetros "aprenden" de los resultados de las capas precedentes. No aprenden directamente de las características de las muestras de datos, que están situadas en la primera capa. Las arquitecturas deep learning (aprendizaje profundo) pueden conformarse a base de varias capas con métodos de machine learning iguales o diferentes, aunque de manera habitual están basadas en redes neuronales multi-capa.

1.2 TRATANDO CON DATOS

En machine learning, los datos son la base de todo; no habrá aprendizaje si no hay suficientes datos, o éstos no son representativos o presentan información sesgada. Cuando la cantidad de datos es insuficiente, los algoritmos de machine learning no pueden generalizar los resultados: simplemente aprenden los patrones específicos de las muestras existentes. Si un niño solo ha visto cinco vehículos a motor, podrá reconocerlos, pero probablemente no podrá generalizarlo y clasificar como vehículo a motor los diferentes tipos de coches, camiones, motos, etc. Este concepto es muy importante en machine learning: se denomina *sobreajuste* (*overfitting*) y los ingenieros de datos deben prevenirlo.

Incluso si disponemos de suficiente cantidad de datos, éstos podrían no ser aceptables para algunos propósitos específicos de machine learning si no son representativos o están sesgados. Como ejemplo: no podremos predecir qué canción le gustará escuchar a una mujer de edad avanzada que acabe de conectarse a un servicio de música online, sobre todo si este servicio es usado normalmente por personas jóvenes: simplemente, los datos están sesgados por las preferencias musicales de los jóvenes. Sin embargo, en este contexto, machine learning puede hacer un gran trabajo recomendando música a usuarios jóvenes. De la misma manera, si ajustamos un modelo para clasificar imágenes de perros y gatos, probablemente funcionará adecuadamente clasificando perros y gatos, pero no podemos esperar que clasifique correctamente leones o tigres: es más, si casi todas las imágenes son de perros pastores alemanes, será muy difícil que clasifique de manera adecuada a los perros chihuahua. Esto es debido a que los datos están sesgados hacia los perros pastores alemanes.

Incluso si vamos a usar algún conjunto de datos que contiene información representativa y no sesgada, machine learning podría presentar fallos si la información no es de calidad. Ejemplos de información de mala calidad son:

- Cuando hay muchas muestras con valores vacíos en alguna característica (*feature*): por ejemplo, personas que no rellenan su edad, o falta el código postal en los formularios, etc.
- Valores atípicos (*outliers*): datos incorrectos provenientes de errores humanos, sensores de IoT que funcionan incorrectamente, errores en los programas: p. ej. Mezcla de medidas entre sistema métrico internacional y sistema métrico anglosajón.
- Datos incorrectos e inconsistentes: direcciones de correo sin el símbolo @, direcciones postales sin el número del portal, nombres de calles que no se corresponden con el código postal, etc.

También las características irrelevantes pueden estropear un proceso de machine learning. Para poder obtener resultados adecuados necesitamos datos relevantes. Si solo se recogen datos basados en características psicológicas será muy difícil predecir la existencia de un tumor cancerígeno. Lo mismo puede ocurrir si se predice la venta de pañales en base a información del tiempo atmosférico.

Cuando se trabaja en el campo de machine learning debemos diferenciar entre características de tipo continuo y características de tipo categórico (*categorical*) (características discretas). Habitualmente resulta equivalente a la división previamente presentada: clasificación vs. regresión, pero aplicado a los datos de entrada. Ejemplos de características de entrada continuas son: la intensidad de color en un pixel, la presión en un sensor, el tiempo de ejecución, precios, etc. Las características categóricas clasifican muestras en grupos: sexo, color (negro, rojo,…), departamento (música, deportes, etc.), país.

Para representar valores categóricos se usa, normalmente, el proceso denominado *codificación one-hot* (one-hot encoding). Básicamente, la codificación one-hot representa una variable categórica mediante el uso de varias características binarias nuevas (una nueva característica por cada valor categórico existente). Imaginemos un sistema de control de incendios en el que tenemos una característica categórica denominada 'riesgo' que solamente admite los valores discretos: riesgo = {'rojo', 'naranja', 'amarillo', 'verde'}. Podríamos tener miles de muestras de edificios con, probablemente, docenas de características, una de las cuales sería 'riesgo'. El ingeniero de datos sabe que la mayoría de los algoritmos de machine learning no admiten valores de entrada categóricos, así que podría transformar los colores que indican el riesgo en valores numéricos que pudiesen alimentar al algoritmo: riesgo = {1,2,3,4}. Puede funcionar, pero resultará mucho más difícil para los algoritmos relacionar los niveles de riesgo con otras características importantes o representativas tales como 'instalaciones de acceso'. La codificación one-hot hace más fácil para los métodos de machine learning la extracción de patrones y la relación entre características, con el objeto de predecir objetivos (tales como 'probabilidad de éxito en una acción de rescate'). En este ejemplo, la característica de riesgo estará codificada one-hot en las siguientes nuevas características binarias: 'riesgoRojo', riesgoNaranja', 'riesgoAmarillo', 'riesgoVerde'.

El proceso denominado *binning (bucketing)* convierte una característica numérica en varias características binarias. Considérese, por ejemplo, una situación en la que la edad de cada uno de los trabajadores determina, en algún grado, la productividad de la empresa: consideremos, en este ejemplo, que los trabajadores más jóvenes y los más mayores serán menos productivos, mientras que la productividad alcanzará su máximo nivel en las edades intermedias. En este escenario, la característica 'edad' no se ajustará a la productividad objetivo; resultaría mucho más

fácil para los algoritmos de machine learning procesar la edad si estuviese codificada de manera categórica (enumerada). Podríamos reemplazar la característica numérica 'edad' por los valores binarios enumerados: 'muyJoven', 'joven', 'edadMedia', 'senior'.

Muchos de los algoritmos de machine learning funcionan mejor cuando todas las características y el valor numérico objetivo están en el mismo rango. Además, evitando que los valores de entrada sean muy grandes se ayuda a los algoritmos a encontrar la solución y a encontrarla de manera más rápida. Por esta razón, de manera habitual, es necesario llevar a cabo un proceso de *Normalización* sobre las muestras. La normalización más simple consiste en dividir cada muestra por el valor máximo de dichas muestras. En este caso, el rango de datos presenta como límite el valor uno. -128, 0, 128, 240, 255 = -0.5, 0.0, 0.5, 0.94, 1. Para obtener un rango que vaya de 0 a 1, la ecuación de normalización es:

$$x_i = \frac{x_i - min(X)}{max(X) - min(X)}$$

La normalización puede fallar cuando hay información sesgada o existen valores atípicos (outliers). Si un valor atípico se cuela entre los datos de entrada, el efecto de la normalización se deteriorará: -128, 0, 128, 2000, 240, 255 -> -0.064, 0.0, 0.064, 1, 0.12, 0.127. Por este motivo, la *estandarización* (*normalización* z-*score o standardization*) es, muy a menudo, la mejor opción. La estandarización centra los datos en su media y los distribuye de acuerdo con el valor de la desviación típica. Un valor estandarizado puede ser interpretado como el número de desviaciones típicas que lo separa de la media.

$$x_i = \frac{x_i - \mu(X)}{\sigma(X)}$$

Para tratar con los valores vacíos en las características, los enfoques típicos son:

- Eliminar las muestras que contengan valores vacíos.
- Insertar los valores correctos cuando sea posible: p.ej.: código postal o dirección postal.
- Utilización de la técnica de *data imputation*.

La técnica de *data imputation* consiste en reemplazar los valores vacíos en una característica por alguna predicción. El sistema más simple para predecir es la media de los valores de la característica: p.ej.: insertar la edad media en cada valor de edad que esté vacío. Un enfoque muy diferente es insertar un valor fuera del rango: p.ej.: una edad de 500. De esta manera, el método de machine learning puede aprender que las muestras con este valor no contribuyen al modelo. De manera similar, se puede añadir una característica nueva para marcar los valores vacíos con el número uno y los no vacíos con el número cero. Finalmente, se puede usar una regresión para predecir los valores.

1.3 MEDICIÓN DE LA CALIDAD

Los ingenieros de datos emplean una gran cantidad de tiempo mejorando la calidad de los resultados de los modelos obtenidos. Normalmente no es posible saber visualmente si un resultado es mejor que otro; resulta necesario procesar algunas medidas de calidad para conocer la exactitud de los resultados. Dependiendo del enfoque de machine learning que estemos usando, podemos emplear diferentes medidas de calidad: algunas son aplicables a resultados de clasificación, otras pueden ser aplicadas a regresión, etc. Vamos a explicar cada una de las medidas de calidad más importantes, pero primero expondremos la metodología.

La regla de oro en metodología de la validación es: no medir la calidad con el mismo conjunto de datos empleado para obtener el modelo. Queremos evitar situaciones en las que el modelo aprende cada una de las muestras de datos, pero es incapaz de tratar datos nuevos. Esta es la situación de sobreajuste (overfitting) que hemos presentado anteriormente, al principio de la sección precedente (pastores alemanes y gatos). Es necesario dividir las muestras de datos y los valores objetivo en varios conjuntos:

- Entrenamiento
- Validación
- Test

El conjunto de datos de entrenamiento es el mayor: un valor típico es que sea el 80% del conjunto total de muestras. Se usa para entrenar el modelo. El conjunto de validación contiene las muestras usadas para mejorar el modelo a base de realizar un ajuste fino en los híper-parámetros (valores que controlan diferentes variaciones en el funcionamiento de los algoritmos). Por último, el conjunto de test (o pruebas) nos permitirá medir la calidad del modelo utilizando las muestras que no hayan sido usadas para entrenar el modelo o para mejorarlo. Debemos asegurarnos de que los tres conjuntos tienen una distribución similar de los datos, que contienen

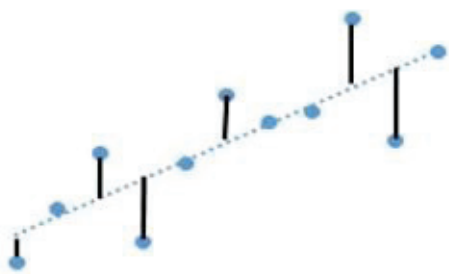
datos representativos y que su intersección es nula. Una vez que hayamos entrenado el modelo con el conjunto de muestras de entrenamiento y lo hayamos mejorado usando el conjunto de muestras de validación, podremos usar las muestras de test para hacer predicciones con estos datos, que no han sido previamente procesados. Los valores indicadores de la calidad serán tanto mayores cuanto más se asemejen las predicciones obtenidas a los valores objetivo existentes.

Dos medidas de calidad bien conocidas son el *Error Medio Absoluto (Mean Absolute Error o MAE) y el Error Cuadrático Medio (Mean Squared Differences o MSD)*. Ambas medidas de calidad penalizan la distancia que hay entre cada valor de una predicción y el valor objetivo. Sus ecuaciones son:

$$MSD(X,y) = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i - p(X_i)\right)^2$$

$$MAE(X,y) = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{y}_i - p(X_i)\right|$$

Ambas ecuaciones devuelven el error medio cometido en las N muestras de test. El error en ambas ecuaciones es la diferencia entre el valor objetivo real y el valor predicho para la muestra X. MSD penaliza los errores con valores grandes (los eleva al cuadrado) mucho más de lo que lo hace el MAE. La siguiente imagen muestra los errores en un modelo de regresión lineal: las líneas verticales que unen los valores reales con los predichos.



Mientras que las medidas anteriores devuelven valores absolutos, la ecuación del coeficiente R^2 (*R^2 score*) devuelve valores relativos. Esta medida de calidad proporciona un valor entre 0 y 1, donde el 1 significa una predicción perfecta, y 0 se corresponde al modelo de regresión más simple: predecir la media del conjunto de muestras de test. Asumiremos que el regresor de prueba no será peor que el regresor más simple. Al coeficiente R^2 se le denomina también *Coeficiente de Determinación*. La ecuación de R^2 es:

$$R^2 = 1 - \frac{\sum_{i=0}^{n-1}(y_i - f(x_i))^2}{\sum_{i=0}^{n-1}(y_i - \overline{y})^2}$$

Donde $f(x)_i$ es la predicción del regresor para la muestra x_i

Los algoritmos de clasificación también hacen predicciones: "esta muestra pertenece a este grupo o a este otro grupo". Las medidas de calidad de regresión podrían utilizarse también en clasificación: asignando ceros a los errores de las muestras clasificadas correctamente y asignando unos a los errores de las muestras clasificadas erróneamente. Existe un enfoque diferente que hace posible diseñar medidas de calidad más específicas para la clasificación. Si centramos nuestra atención en la clasificación binaria, en la matriz de confusión solamente pueden existir dos clases (grupos):

Predicción	clase 0 (positivo)	clase 1 (negativo)
clase 0	verdadero positivo (TP)	falso negativo (FN)
clase1	falso positivo (FP)	verdadero negativo (TN)

Para mostrar un ejemplo ilustrativo, supongamos que tenemos 100 imágenes de test de radiografías etiquetadas: ('tumor maligno', 'tumor benigno'). 35 de ellas han sido predichas correctamente como tumores benignos, 45 han sido predichas correctamente como malignos, 5 han sido predichas incorrectamente como benignos y 15 han sido incorrectamente predichas como malignos

Predicción Tumor	Benigno (+)	Maligno (-)
Benigno	35	15
Maligno	5	45

La medida de calidad denominada *Precision* nos muestra la proporción de aciertos en la predicción: ¿Cuántas de sus predicciones son correctas? La medida de calidad *Recall* centra su atención en el número relativo de muestras positivas (en nuestro ejemplo, tumores benignos): ¿Cuántas de sus predicciones son correctas de manera relativa al número total de muestras positivas? Mientras que *Precision* nos proporciona un valor de calidad relativo al número total de predicciones realizadas, *Recall* nos proporciona un valor de calidad relativo al número total de muestras positivas. Merece la pena mantener niveles altos de *Precision* cuando el número

de predicciones crece. Merece la pena mantener niveles altos de *Recall* cuando el número de muestras positivas es bajo. Dependiendo de la semántica de cada problema de inteligencia artificial, los valores de la matriz de confusión cobran mayor o menor importancia. En nuestro ejemplo es vital mantener el número de falsos positivos (tumores malignos no detectados) tan bajo como sea posible y resulta muy importante predecir de manera correcta los negativos verdaderos (tumores malignos detectados).

$$Precision = \frac{TP}{TP + FP}$$
$$Recall = \frac{TP}{TP + FN}$$

Existe una medida de calidad de clasificación que une Precision y Recall; es la *F1*:

$$F1 = 2\frac{Precision \times Recall}{Precision + recall}$$

En el caso de que todas las clases sean igualmente importantes se puede usar la siguiente medida de calidad de exactitud (accuracy):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

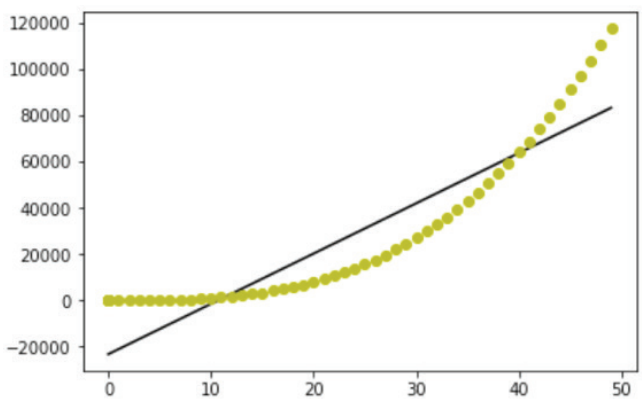
Para medir la calidad de la clasificación también se usa la curva ROC: combina los valores verdaderos positivos con los falsos positivos, definida como:

$$TPR = \frac{TP}{TP + FN}, \qquad FPR = \frac{FP}{FP + TN}$$

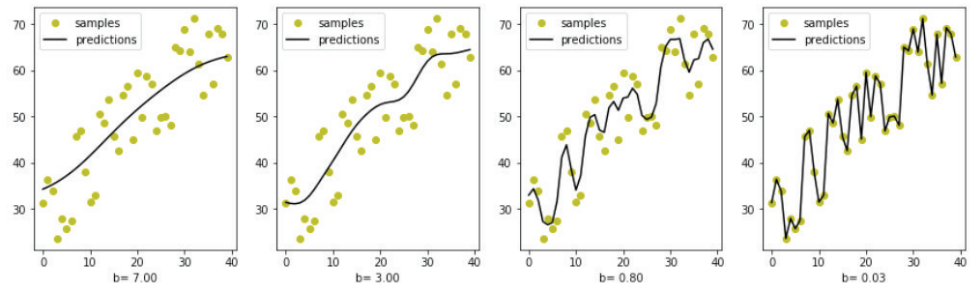
Cuanto mayor sea el área bajo la curva ROC, mejor será el modelo. Los valores obtenidos deben ser mayores de 0.5, ya que 0.5 se corresponde con un clasificador aleatorio.

1.4 MEJORA DEL MODELO

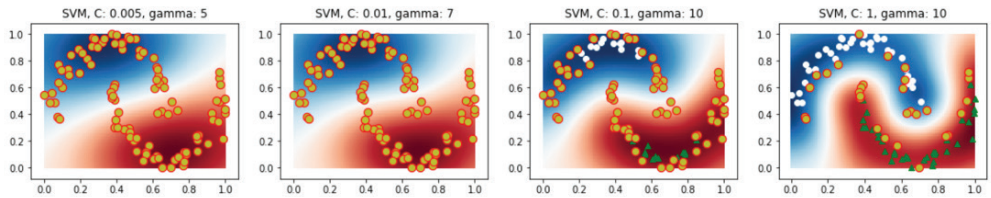
Para mejorar un modelo de machine learning se debe medir la calidad que obtiene al actuar sobre el conjunto de muestras de test (pruebas). En esta sección nos centramos en los conjuntos de datos de entrenamiento y de pruebas (test), pero debemos recordar que los proyectos profesionales de machine learning deben usar también el conjunto de datos de validación. Primero se usa el conjunto de datos de entrenamiento para crear el modelo, y podemos, entonces, obtener sus medidas de calidad (aplicadas a las muestras de entrenamiento). Si las predicciones del modelo presentan muchos errores, las medidas de calidad devolverán valores muy bajos. Esto significa que los datos no son suficientemente numerosos, o presentan muchos valores atípicos, o las características no son relevantes, o que el algoritmo de aprendizaje utilizado no está indicado para la presente tarea (normalmente por ser muy simple), o los híper-parámetros de aprendizaje elegidos no son correctos. En estos casos, el modelo sufre *subajuste* (*underfitting*). Para ilustrar el caso, reproducimos la siguiente figura, obtenida de datos generados. Se puede ver que el modelo lineal es demasiado simple para ajustar la distribución no lineal de los datos.



La situación opuesta al subajuste es la del sobreajuste (overfitting). Un modelo presenta sobreajuste cuando predice con suficiente exactitud el conjunto de datos de entrenamiento y, sin embargo, no puede predecir adecuadamente las muestras de test: no generaliza. La siguiente figura visualiza varios modelos obtenidos a partir de entrenamientos que utilizan el mismo conjunto de muestras. Los dos gráficos de la izquierda representan modelos más simples, que son capaces de generalizar para predecir de manera adecuada nuevas muestras. Los dos gráficos de la derecha representan modelos más complejos, que no son capaces de generalizar: presentan situaciones de sobreajuste.



Para evitar el sobreajuste podemos entrenar el modelo usando más datos: en este caso será más difícil para el algoritmo ajustar todos los datos. Por supuesto, normalmente no disponemos de más datos. Otra posibilidad es usar un modelo más simple (quizás un modelo lineal). Reducir la dimensionalidad de las muestras también puede funcionar: puede ser llevado a cabo usando técnicas de reducción de dimensiones. Por último, *regularizar* el modelo es, normalmente, la solución más simple y más eficiente. La *regularización* se puede llevar a cabo de diferentes maneras, dependiendo de cada algoritmo de machine learning que se esté usando, pero su concepto subyacente es siempre simplificar el modelo. A menudo, la regularización significa mantener los pesos de aprendizaje tan pequeños como sea posible: limita la importancia de algunas características, simplificando el modelo. En algunos casos, la regularización reduce en algún grado el ámbito de aprendizaje: p.ej.: el número máximo de niveles que va a tener un árbol de decisión. La siguiente figura muestra los efectos de la regularización; en este caso, el parámetro *C* determina la flexibilidad permitida para crear el modelo. Cuando el valor de *C* es alto, el modelo es complicado y sufre sobreajuste. Los valores bajos de *C* limitan la flexibilidad del aprendizaje y simplifican el modelo: contribuyen a generalizar y a obtener mayor exactitud.



Otra técnica específica de regularización es la denominada *data augmentation* (enriquecimiento de datos). Ya que incrementar el número de muestras de entrenamiento reduce el sobreajuste, podemos generar nuevas muestras de entrenamiento a base de combinar las existentes. Aunque la técnica de *data augmentation* puede ser usada, teóricamente, en cualquier ámbito, se aplica especialmente en clasificación de imágenes. Se pueden crear nuevas imágenes simplemente rotando, escalando, combinando, etc. las muestras de entrenamiento existentes.