Prefacio

El tsunami del aprendizaje automático

En 2006, Geoffrey Hinton et al. publicó un artículo [1] mostrando cómo entrenar una red neuronal profunda capaz de reconocer dígitos escritos a mano con precisión de vanguardia (> 98%). Llamaron a esta técnica "Aprendizaje profundo". Entrenar una red neuronal profunda se consideraba imposible en ese momento, [2] y la mayoría de los investigadores habían abandonado la idea desde la década de 1990. Este artículo reavivó el interés de la comunidad científica y, en poco tiempo, muchos artículos nuevos demostraron que el aprendizaje profundo no solo era posible, sino que también era capaz de lograr logros alucinantes que ninguna otra técnica de aprendizaje automático (ML) podría esperar igualar (con la ayuda de tremendos potencia de cálculo y grandes cantidades de datos). Este entusiasmo pronto se extendió a muchas otras áreas del aprendizaje automático.

Avance rápido 10 años y el aprendizaje automático ha conquistado la industria: ahora está en el corazón de gran parte de la magia de los productos de alta tecnología actuales, clasificando los resultados de búsqueda web, potenciando el reconocimiento de voz de su teléfono inteligente y recomendando videos, superando al mundo campeón en el juego de Go. Antes de que te des cuenta, estará conduciendo tu coche.

Aprendizaje automático en sus proyectos

¡Así que, naturalmente, estás entusiasmado con el aprendizaje automático y te encantaría unirte a la fiesta!

¿Quizás le gustaría darle a su robot casero un cerebro propio? ¿Hacer que reconozca caras? ¿O aprender a caminar?

1 Disponible en la página de inicio de Hinton en *http://www.cs.toronto.edu/~hinton/* .

2 A pesar del hecho de que las profundas redes neuronales convolucionales de Yann Lecun habían funcionado bien para el reconocimiento de imágenes desde la década de 1990, aunque no tenían un propósito tan general.

O tal vez su empresa tiene toneladas de datos (registros de usuarios, datos financieros, datos de producción, datos de sensores de máquinas, estadísticas de línea directa, informes de recursos humanos, etc.), y lo más probable es que pueda descubrir algunas gemas ocultas si supiera dónde buscar; por ejemplo:

- Segmentar clientes y encontrar la mejor estrategia de marketing para cada grupo
- Recomendar productos para cada cliente en función de lo que compraron clientes similares.
- Detectar las transacciones que probablemente sean fraudulentas
- Predecir los ingresos del próximo año
- Y más

Cualquiera sea el motivo, ha decidido aprender Machine Learning e implementarlo en sus proyectos. ¡Gran idea!

Objetivo y enfoque

Este libro asume que no sabe casi nada sobre el aprendizaje automático. Su objetivo es brindarle los conceptos, las intuiciones y las herramientas que necesita para implementar programas capaces de *aprender de los datos*.

Cubriremos una gran cantidad de técnicas, desde las más simples y de uso más común (como la regresión lineal) hasta algunas de las técnicas de Deep Learning que suelen ganar concursos.

En lugar de implementar nuestras propias versiones de juguete de cada algoritmo, usaremos marcos de Python listos para producción reales:

- Scikit-Learn Es muy fácil de usar, pero implementa muchos algoritmos de Machine Learning de manera eficiente, por lo que es un excelente punto de entrada para aprender Machine Learning.
- TensorFlow es una biblioteca más compleja para el cálculo numérico distribuido utilizando gráficos de flujo de datos. Hace posible entrenar y ejecutar redes neuronales muy grandes de manera eficiente al distribuir los cálculos a través de potencialmente miles de servidores multi-GPU. TensorFlow se creó en Google y es compatible con muchas de sus aplicaciones de aprendizaje automático a gran escala. Fue de código abierto en noviembre de 2015.

El libro favorece un enfoque práctico, aumentando una comprensión intuitiva del aprendizaje automático a través de ejemplos de trabajo concretos y solo un poco de teoría. Si bien puede leer este libro sin levantar su computadora portátil, le recomendamos que experimente con los ejemplos de código disponibles en línea como cuadernos de Jupyter en *https://github.com/ageron/handson-ml* .

Prerrequisitos

Este libro asume que tiene algo de experiencia en programación Python y que está familiarizado con las principales bibliotecas científicas de Python, en particular NumPy , Pandas y Matplotlib .

Además, si le importa lo que hay debajo del capó, también debe tener una comprensión razonable de las matemáticas de nivel universitario (cálculo, álgebra lineal, probabilidades y estadísticas).

Si aún no conoce Python, *http://learnpython.org/* es un gran lugar para comenzar. El tutorial oficial sobre python.org también es bastante bueno.

Si nunca ha utilizado Jupyter, Capitulo 2 lo guiará a través de la instalación y los conceptos básicos: es una gran herramienta para tener en su caja de herramientas.

Si no está familiarizado con las bibliotecas científicas de Python, los cuadernos de notas de Jupyter proporcionados incluyen algunos tutoriales. También hay un tutorial rápido de matemáticas para álgebra lineal.

Mapa vial

Este libro está organizado en dos partes. Parte I *Los fundamentos del aprendizaje automático ,* cubre los siguientes temas:

- ¿Qué es el aprendizaje automático? ¿Qué problemas intenta resolver? ¿Cuáles son las principales categorías y conceptos fundamentales de los sistemas de Machine Learning?
- Los pasos principales en un proyecto típico de Machine Learning.
- Aprender ajustando un modelo a los datos.
- Optimización de una función de costes.
- Manejo, limpieza y preparación de datos.
- Selección e ingeniería de características.
- Seleccionar un modelo y ajustar los hiperparámetros mediante validación cruzada.
- Los principales desafíos del aprendizaje automático, en particular el desajuste y el sobreajuste (el equilibrio entre sesgo y varianza).
- Reducir la dimensionalidad de los datos de entrenamiento para luchar contra la maldición de la dimensionalidad.
- Los algoritmos de aprendizaje más comunes: regresión lineal y polinomial, regresión logística, k-vecinos más cercanos, máquinas de vectores de soporte, árboles de decisión, bosques aleatorios y métodos de conjunto.

Parte II, *Redes neuronales y aprendizaje profundo* , cubre los siguientes temas:

- ¿Qué son las redes neuronales? ¿Para qué son buenos?
- Construir y entrenar redes neuronales con TensorFlow.
- Las arquitecturas de redes neuronales más importantes: redes neuronales de retroalimentación, redes convolucionales, redes recurrentes, redes de memoria a largo plazo a corto plazo (LSTM) y autocodificadores.
- Técnicas de entrenamiento de redes neuronales profundas.
- Escalar redes neuronales para grandes conjuntos de datos.
- Aprendizaje reforzado.

La primera parte se basa principalmente en Scikit-Learn, mientras que la segunda parte usa TensorFlow.

No te adentres demasiado en aguas profundas: aunque el aprendizaje profundo es sin duda una de las áreas más interesantes del aprendizaje automático, primero debes dominar los fundamentos. Además, la mayoría de los problemas se pueden resolver bastante bien utilizando técnicas más simples como los métodos de bosques aleatorios y conjuntos (discutidos en Parte I). El aprendizaje profundo es más adecuado para problemas complejos como el reconocimiento de imágenes, el reconocimiento de voz o el procesamiento del lenguaje natural, siempre que tenga suficientes datos, poder de cómputo y paciencia.

Otros recursos

Hay muchos recursos disponibles para aprender sobre el aprendizaje automático. Andrew Ng Curso de aprendizaje automático en Coursera y Geoffrey Hinton's curso sobre redes neuronales y Deep Learning son increíbles, aunque ambos requieren una inversión de tiempo significativa (piense en meses).

También hay muchos sitios web interesantes sobre Machine Learning, incluido, por supuesto, el excepcional software de Scikit-Learn. Guía del usuario . También puede disfrutar Dataquest , que proporciona tutoriales interactivos muy agradables y blogs ML como los que se enumeran en Quora . Finalmente, el Sitio web de Deep Learning tiene una buena lista de recursos para aprender más.

Por supuesto, también hay muchos otros libros introductorios sobre Machine Learning, en particular:

- Joel Grus, *Ciencia de datos desde cero (* O'Reilly). Este libro presenta los fundamentos del aprendizaje automático e implementa algunos de los principales algoritmos en Python puro (desde cero, como su nombre indica).
- Stephen Marsland, *Aprendizaje automático: una perspectiva algorítmica (* Chapman y Hall). Este libro es una excelente introducción al aprendizaje automático, que cubre una amplia

variedad de temas en profundidad, con ejemplos de código en Python (también desde cero, pero usando NumPy).

- Sebastian Raschka, *Aprendizaje automático de Python (* Packt Publishing). También es una gran introducción al aprendizaje automático, este libro aprovecha las bibliotecas de código abierto de Python (Pylearn 2 y Theano).

- Yaser S. Abu-Mostafa, Malik Magdon-Ismail y Hsuan-Tien Lin, *Aprender de los datos (* AMLBook). Este libro, que es un enfoque bastante teórico del ML, proporciona conocimientos profundos, en particular sobre el equilibrio entre sesgo y varianza (ver Capítulo 4).

- Stuart Russell y Peter Norvig, *Inteligencia artificial: un enfoque moderno, tercera edición (* Pearson). Este es un gran (y enorme) libro que cubre una cantidad increíble de temas, incluido el aprendizaje automático. Ayuda a poner el aprendizaje automático en perspectiva.

Finalmente, una excelente manera de aprender es unirse a los sitios web de competencia de ML como Kaggle.com esto le permitirá practicar sus habilidades en problemas del mundo real, con la ayuda y los conocimientos de algunos de los mejores profesionales de ML que existen.

Las convenciones usadas en este libro

En este libro se utilizan las siguientes convenciones tipográficas:

Itálico

Indica nuevos términos, URL, direcciones de correo electrónico, nombres de archivo y extensiones de archivo.

Ancho constante

Se utiliza para listados de programas, así como dentro de párrafos para referirse a elementos de programa como nombres de variables o funciones, bases de datos, tipos de datos, variables de entorno, declaraciones y palabras clave.

Ancho constante en negrita

Muestra comandos u otro texto que el usuario debe escribir literalmente.

Cursiva de ancho constante

Muestra texto que debe reemplazarse por valores proporcionados por el usuario o por valores determinados por contexto.

Este elemento significa un consejo o sugerencia.

Este elemento significa una nota general.

Este elemento indica una advertencia o precaución.

Usar ejemplos de código

El material complementario (ejemplos de código, ejercicios, etc.) está disponible para descargar en *https://github.com/ageron/handson-ml* .

Este libro está aquí para ayudarlo a hacer su trabajo. En general, si se ofrece un código de ejemplo con este libro, puede utilizarlo en sus programas y documentación. No es necesario que se comunique con nosotros para obtener permiso a menos que esté reproduciendo una parte significativa del código. Por ejemplo, escribir un programa que utiliza varios fragmentos de código de este libro no requiere permiso. Vender o distribuir un CD-ROM de ejemplos de libros de O'Reilly requiere permiso. Responder una pregunta citando este libro y citando un código de ejemplo no requiere permiso. La incorporación de una cantidad significativa de código de ejemplo de este libro en la documentación de su producto requiere permiso.

Agradecemos la atribución, pero no la exigimos. Una atribución generalmente incluye el título, el autor, el editor y el ISBN. Por ejemplo: " *Aprendizaje automático práctico con Scikit-Learn y TensorFlow* de Aurélien Géron (O'Reilly). Copyright 2017 Aurélien Géron, 978-1-491-96229-9 ".

Si cree que el uso que hace de los ejemplos de código está fuera del uso legítimo o del permiso otorgado anteriormente, no dude en contactarnos en *permissions@oreilly.com* .

Safari de O'Reilly

Safari (anteriormente Safari Books Online) es una plataforma de referencia y capacitación basada en membresías para empresas, gobiernos, educadores e individuos.

Los miembros tienen acceso a miles de libros, videos de capacitación, rutas de aprendizaje, tutoriales interactivos y listas de reproducción seleccionadas de más de 250 editores, incluidos O'Reilly Media, Harvard Business Review, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press , Sams, Que, Peachpit Press, Adobe, Focal Press, Cisco Press,

John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett y Course Technology, entre otros.

Para mayor información por favor visite *http://oreilly.com/safari* .

Cómo contactarnos

Dirija sus comentarios y preguntas sobre este libro al editor:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (en los Estados Unidos o Canadá)
707-829-0515 (internacional o local)
707-829-0104 (fax)

Tenemos una página web para este libro, donde enumeramos erratas, ejemplos y cualquier información adicional. Puede acceder a esta página en *http://bit.ly/hands-on-machine-learning- with-scikit-learn-and-tensorflow* .

Para comentar o hacer preguntas técnicas sobre este libro, envíe un correo electrónico a *bookques- tions@oreilly.com* .

Para obtener más información sobre nuestros libros, cursos, conferencias y noticias, visite nuestro sitio web en *http://www.oreilly.com* .

Encuentranos en Facebook: *http://facebook.com/oreilly*

Síguenos en Twitter: *http://twitter.com/oreillymedia*

Míranos en YouTube: *http://www.youtube.com/oreillymedia*

Expresiones de gratitud

Me gustaría agradecer a mis colegas de Google, en particular al equipo de clasificación de videos de YouTube, por enseñarme tanto sobre el aprendizaje automático. Nunca podría haber comenzado este proyecto sin ellos. Un agradecimiento especial a mis gurús personales del aprendizaje automático: Clément Courbet, Julien Dubois, Mathias Kende, Daniel Kitachewsky, James Pack, Alexander Pak, Anosh Raj, Vitor Sessak, Wiktor Tomczak, Ingrid von Glehn, Rich Washington y todos en YouTube París.

Estoy increíblemente agradecido con todas las personas increíbles que se tomaron un tiempo de sus ocupadas vidas para revisar mi libro con tanto detalle. Gracias a Pete Warden por responder todas mis preguntas de TensorFlow, revisar Parte II , proporcionando muchas ideas interesantes y, por supuesto, por ser parte del equipo central de TensorFlow. Definitivamente deberías echarle un vistazo

su blog ! Muchas gracias a Lukas Biewald por su minuciosa revisión de Parte II : no dejó piedra sin remover, probó todo el código (y detectó algunos errores), hizo muchas sugerencias excelentes y su entusiasmo fue contagioso. Deberías echar un vistazo su blog y su robots geniales ! Gracias a Justin Francis, quien también revisó Parte II muy a fondo, detectando errores y proporcionando grandes conocimientos, en particular en Capítulo 16 . Echa un vistazo

sus publicaciones en TensorFlow!

Muchas gracias también a David Andrzejewski, quien revisó Parte I y proporcionó comentarios increíblemente útiles, identificando secciones poco claras y sugiriendo cómo mejorarlas. Echa un vistazo su sitio web ! Gracias a Grégoire Mesnil, que revisó
Parte II y contribuyó con consejos prácticos muy interesantes sobre el entrenamiento de redes neuronales. Gracias también a Eddy Hung, Salim Sémaoune, Karim Matrah, Ingrid von Glehn, Iain Smears y Vincent Guilbeau por revisar Parte I y haciendo muchas sugerencias útiles. Y también deseo agradecer a mi suegro, Michel Tessier, ex profesor de matemáticas y ahora un gran traductor de Anton Chejov, por ayudarme a pulir algunas de las matemáticas y notaciones en este libro y revisar el álgebra lineal Jupyter cuaderno.

Y, por supuesto, un gigantesco "gracias" a mi querido hermano Sylvain, quien revisó cada capítulo, probó cada línea de código, brindó comentarios sobre prácticamente cada sección y me animó desde la primera línea hasta la última. ¡Te amo hermano!

Muchas gracias también al fantástico personal de O'Reilly, en particular a Nicole Tache, quien me brindó comentarios profundos, siempre alegres, alentadores y útiles. Gracias también a Marie Beaugureau, Ben Lorica, Mike Loukides y Laurel Ruma por creer en este proyecto y ayudarme a definir su alcance. Gracias a Matt Hacker y a todo el equipo de Atlas por responder a todas mis preguntas técnicas sobre el formato, asciidoc y LaTeX, y gracias a Rachel Monaghan, Nick Adams y todo el equipo de producción por su revisión final y sus cientos de correcciones.

Por último, pero no menos importante, estoy infinitamente agradecido a mi amada esposa, Emmanuelle, y a nuestros tres maravillosos hijos, Alexandre, Rémi y Gabrielle, por animarme a trabajar duro en este libro, haciendo muchas preguntas (quienes dijeron que no puedes ¿Enseñar redes neuronales a un niño de siete años?), e incluso traerme galletas y café. ¿Con qué más se puede soñar?

PARTE I

Los fundamentos de Aprendizaje automático

CAPÍTULO 1

El panorama del aprendizaje automático

Cuando la mayoría de la gente escucha "Machine Learning", se imagina un robot: un mayordomo confiable o un Terminator mortal, según a quién le preguntes. Pero el aprendizaje automático no es solo una fantasía futurista, ya está aquí. De hecho, ha existido durante décadas en algunas aplicaciones especializadas, como *Reconocimiento óptico de caracteres (* LOC). Pero la primera aplicación de ML que realmente se volvió popular, mejorando las vidas de cientos de millones de personas, se apoderó del mundo en la década de 1990: fue la *filtro de spam.*

No es exactamente un Skynet consciente de sí mismo, pero técnicamente califica como aprendizaje automático (en realidad, ha aprendido tan bien que rara vez es necesario marcar un correo electrónico como spam). Le siguieron cientos de aplicaciones de aprendizaje automático que ahora impulsan silenciosamente cientos de productos y funciones que utiliza regularmente, desde mejores recomendaciones hasta búsquedas por voz.

¿Dónde comienza el aprendizaje automático y dónde termina? ¿Qué significa exactamente que una máquina *aprender* ¿alguna cosa? Si descargo una copia de Wikipedia, ¿mi computadora realmente ha "aprendido" algo? ¿Es de repente más inteligente? En este capítulo, comenzaremos aclarando qué es el aprendizaje automático y por qué es posible que desee utilizarlo.

Luego, antes de comenzar a explorar el continente del aprendizaje automático, echaremos un vistazo al mapa y aprenderemos sobre las principales regiones y los puntos de referencia más notables: aprendizaje supervisado versus no supervisado, aprendizaje en línea versus aprendizaje por lotes, basado en instancias versus basado en modelos. aprendizaje. Luego, analizaremos el flujo de trabajo de un proyecto de aprendizaje automático típico, discutiremos los principales desafíos que puede enfrentar y cubriremos cómo evaluar y ajustar un sistema de aprendizaje automático.

Este capítulo presenta muchos conceptos fundamentales (y jerga) que todo científico de datos debería conocer de memoria. Será una descripción general de alto nivel (el único capítulo sin mucho código), todo bastante simple, pero debe asegurarse de que todo sea claro antes de continuar con el resto del libro. ¡Así que tómate un café y comencemos!

Si ya conoce todos los conceptos básicos del aprendizaje automático, es posible que desee pasar directamente a Capitulo 2 . Si no está seguro, intente responder todas las preguntas enumeradas al final del capítulo antes de continuar.

¿Qué es el aprendizaje automático?

El aprendizaje automático es la ciencia (y el arte) de programar computadoras para que puedan *aprender de los datos*.

Aquí hay una definición un poco más general:

> [Machine Learning es el] campo de estudio que brinda a las computadoras la capacidad de aprender sin ser programadas explícitamente.
>
> - Arthur Samuel, *1959*

Y uno más orientado a la ingeniería:

> Se dice que un programa de computadora aprende de la experiencia E con respecto a alguna tarea T y alguna medida de desempeño P, si su desempeño en T, medido por P, mejora con la experiencia E.
>
> - TomMitchell, *1997*

Por ejemplo, su filtro de correo no deseado es un programa de aprendizaje automático que puede aprender a marcar el correo no deseado con ejemplos de correos electrónicos no deseados (p. Ej., Marcados por usuarios) y ejemplos de correos electrónicos regulares (no spam, también denominados "ham"). Los ejemplos que utiliza el sistema para aprender se denominan *conjunto de entrenamiento.* Cada ejemplo de entrenamiento se llama *instancia de entrenamiento* o *muestra).*
En este caso, la tarea T es marcar el spam para nuevos correos electrónicos, la experiencia E es la *datos de entrenamiento,* y es necesario definir la medida de desempeño P; por ejemplo, puede utilizar la proporción de correos electrónicos clasificados correctamente. Esta medida de desempeño en particular se llama
exactitud y se utiliza a menudo en tareas de clasificación.

Si acaba de descargar una copia de Wikipedia, su computadora tiene muchos más datos, pero de repente no es mejor en ninguna tarea. Por lo tanto, no es aprendizaje automático.

¿Por qué utilizar el aprendizaje automático?

Considere cómo escribiría un filtro de spam utilizando técnicas de programación tradicionales (Figura 1-1):

1. En primer lugar, debería ver cómo se ve normalmente el spam. Es posible que observe que algunas palabras o frases (como "4U", "tarjeta de crédito", "gratis" e "increíble") tienden a aparecer mucho en el tema. Quizás también observe algunos otros patrones en el nombre del remitente, el cuerpo del correo electrónico, etc.

2. Escribiría un algoritmo de detección para cada uno de los patrones que notó, y su programa marcaría los correos electrónicos como spam si se detectaran varios de estos patrones.

3. Debería probar su programa y repetir los pasos 1 y 2 hasta que sea lo suficientemente bueno.

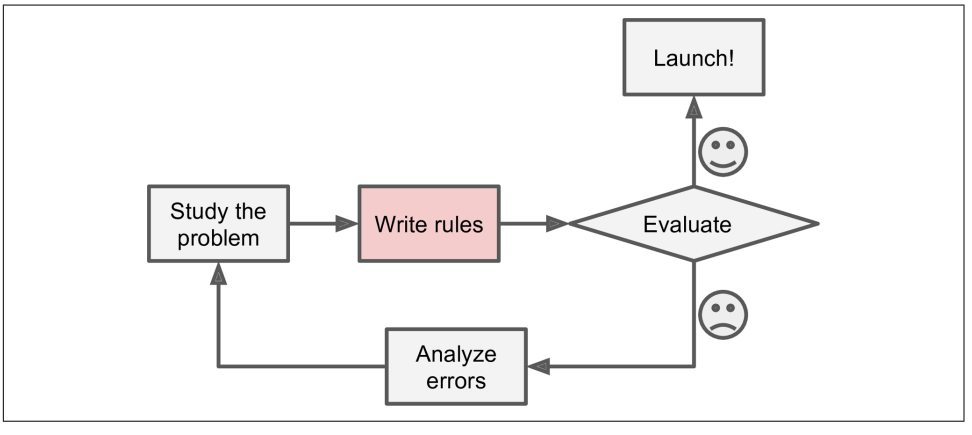

Figura 1-1. El enfoque tradicional

Dado que el problema no es trivial, es probable que su programa se convierta en una larga lista de reglas complejas, bastante difíciles de mantener.

Por el contrario, un filtro de spam basado en técnicas de aprendizaje automático aprende automáticamente qué palabras y frases son buenos predictores de spam al detectar patrones de palabras inusualmente frecuentes en los ejemplos de spam en comparación con los ejemplos de radioaficionados (Figura 1-2). El programa es mucho más corto, más fácil de mantener y probablemente más preciso.

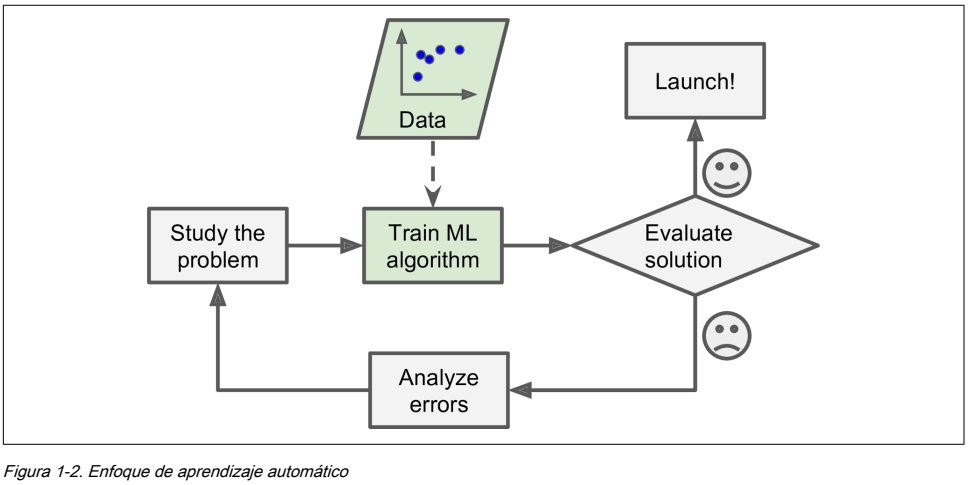

Figura 1-2. Enfoque de aprendizaje automático

Además, si los spammers notan que todos sus correos electrónicos que contienen "4U" están bloqueados, pueden comenzar a escribir "For U" en su lugar. Un filtro de spam que utiliza técnicas de programación tradicionales debería actualizarse para marcar los correos electrónicos "For U". Si los spammers siguen trabajando alrededor de su filtro de spam, deberá seguir escribiendo nuevas reglas para siempre.

Por el contrario, un filtro de spam basado en técnicas de aprendizaje automático notifica automáticamente que "For U" se ha vuelto inusualmente frecuente en el spam marcado por los usuarios, y comienza a marcarlos sin su intervención (Figura 1-3).

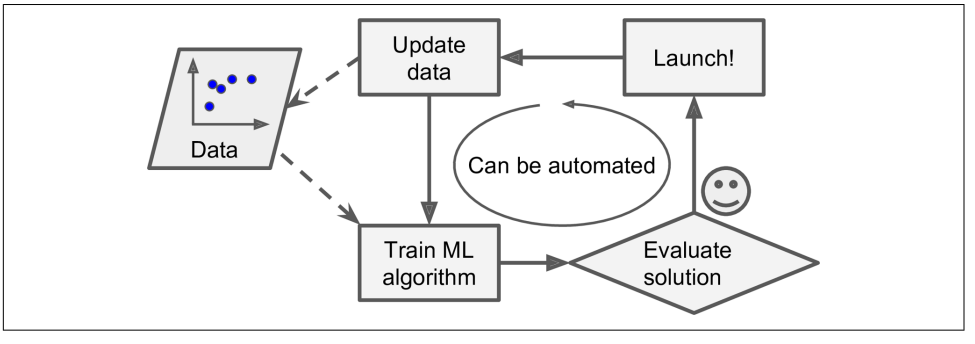

Figura 1-3. Adaptarse automáticamente al cambio

Otra área en la que destaca el aprendizaje automático son los problemas que son demasiado complejos para los enfoques tradicionales o que no tienen un algoritmo conocido. Por ejemplo, considere el reconocimiento de voz: digamos que desea comenzar de manera simple y escribir un programa capaz de distinguir las palabras "uno" y "dos". Es posible que observe que la palabra "dos" comienza con un sonido de tono alto ("T"), por lo que podría codificar un algoritmo que mida la intensidad del sonido de tono alto y usarlo para distinguir unos y dos. Obviamente, esta técnica no se escalará a miles de palabras pronunciadas por millones de personas muy diferentes en entornos ruidosos y en decenas de idiomas. La mejor solución (al menos hoy) es escribir un algoritmo que aprenda por sí mismo, dados muchos ejemplos de grabaciones para cada palabra.

Finalmente, el aprendizaje automático puede ayudar a los humanos a aprender (Figura 1-4): Los algoritmos ML se pueden inspeccionar para ver qué han aprendido (aunque para algunos algoritmos esto puede ser complicado). Por ejemplo, una vez que el filtro de spam ha sido entrenado con suficiente spam, se puede inspeccionar fácilmente para revelar la lista de palabras y combinaciones de palabras que cree que son los mejores predictores de spam. A veces, esto revelará correlaciones insospechadas o nuevas tendencias y, por lo tanto, conducirá a una mejor comprensión del problema.

La aplicación de técnicas de AA para profundizar en grandes cantidades de datos puede ayudar a descubrir patrones que no fueron evidentes de inmediato. Se llama *procesamiento de datos*.

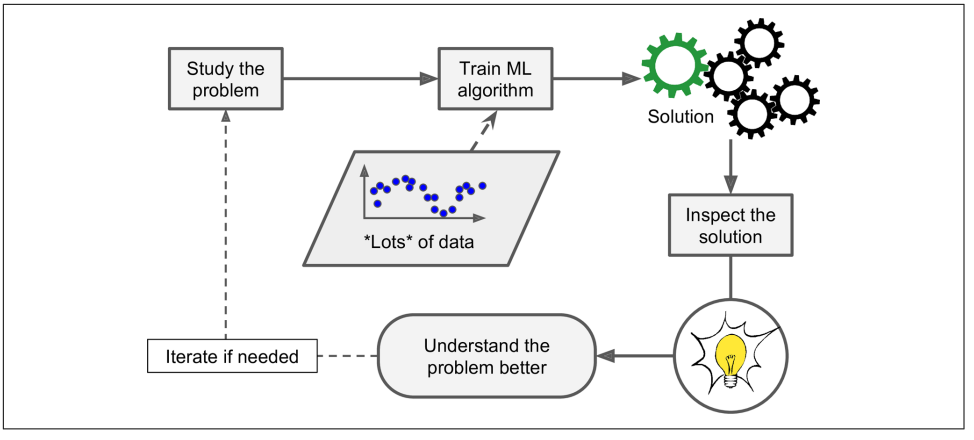

Figura 1-4. El aprendizaje automático puede ayudar a los humanos a aprender

En resumen, el aprendizaje automático es ideal para:

- Problemas para los que las soluciones existentes requieren muchos ajustes manuales o largas listas de reglas: un algoritmo de aprendizaje automático a menudo puede simplificar el código y funcionar mejor.
- Problemas complejos para los que no existe una buena solución utilizando un enfoque tradicional: las mejores técnicas de Machine Learning pueden encontrar una solución.
- Entornos fluctuantes: un sistema de aprendizaje automático puede adaptarse a nuevos datos.
- Obtener conocimientos sobre problemas complejos y grandes cantidades de datos.

Tipos de sistemas de aprendizaje automático

Hay tantos tipos diferentes de sistemas de aprendizaje automático que es útil clasificarlos en categorías amplias según:

- Si están capacitados o no con supervisión humana (supervisados, no supervisados, semisupervisados y aprendizaje reforzado)
- Si pueden aprender o no de forma incremental sobre la marcha (aprendizaje en línea o por lotes)
- Ya sea que funcionen simplemente comparando nuevos puntos de datos con puntos de datos conocidos o que, en su lugar, detecten patrones en los datos de entrenamiento y creen un modelo predictivo, como lo hacen los científicos (aprendizaje basado en instancias versus aprendizaje basado en modelos)

Estos criterios no son exclusivos; puedes combinarlos de la forma que quieras. Por ejemplo, un filtro de spam de última generación puede aprender sobre la marcha utilizando una red neuronal profunda.

modelo de trabajo capacitado utilizando ejemplos de spam y jamón; esto lo convierte en un sistema de aprendizaje supervisado, basado en modelos y en línea.

Veamos cada uno de estos criterios un poco más de cerca.

Aprendizaje supervisado / no supervisado

Los sistemas de aprendizaje automático se pueden clasificar según la cantidad y el tipo de supervisión que reciben durante el entrenamiento. Hay cuatro categorías principales: aprendizaje supervisado, aprendizaje no supervisado, aprendizaje semisupervisado y aprendizaje reforzado.

Aprendizaje supervisado

En *aprendizaje supervisado,* los datos de entrenamiento que alimenta al algoritmo incluyen las soluciones deseadas, llamadas *etiquetas (*Figura 1-5).

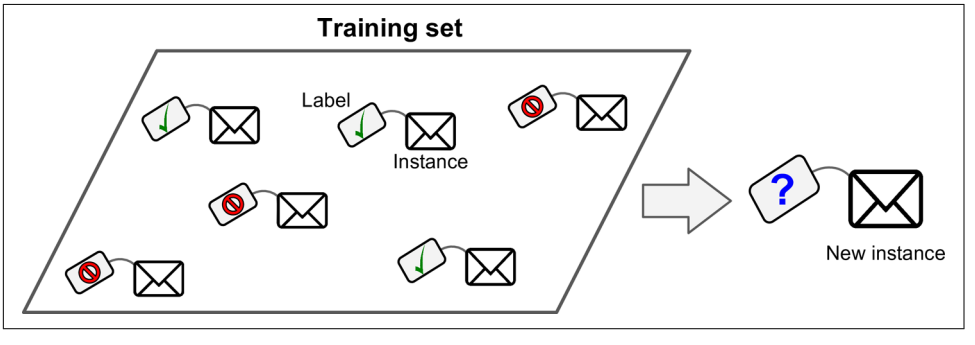

Figura 1-5. Un conjunto de formación etiquetado para el aprendizaje supervisado (p. Ej., Clasificación de spam)

Una tarea típica de aprendizaje supervisado es *clasificación.* El filtro de spam es un buen ejemplo de esto: está entrenado con muchos correos electrónicos de ejemplo junto con sus *clase (* spam o ham), y debe aprender a clasificar nuevos correos electrónicos.

Otra tarea típica es predecir un *objetivo* valor numérico, como el precio de un automóvil, dado un conjunto de *Características (* kilometraje, edad, marca, etc.) llamado *predictores.* Este tipo de tarea se llama *regresión* Figura 1-6). [1] Para entrenar el sistema, debe proporcionarle muchos ejemplos de automóviles, incluidos sus predictores y sus etiquetas (es decir, sus precios).

1 Dato curioso: este nombre que suena extraño es un término estadístico introducido por Francis Galton mientras estudiaba el hecho de que los hijos de personas altas tienden a ser más bajos que sus padres. Como los niños eran más pequeños, llamó a esto *regresión a la media.* Este nombre se aplicó luego a los métodos que usó para analizar las correlaciones entre variables.