

La etapa de entrenamiento de un modelo de machine learning

La etapa de entrenamiento de un modelo de machine learning es crucial y abarca varios pasos esenciales para desarrollar un modelo capaz de realizar predicciones o clasificaciones basadas en datos. Aquí te explico los aspectos fundamentales de este proceso:

1. Preparación de los Datos

Antes de entrenar el modelo, los datos deben ser recopilados, limpiados y organizados. Esto incluye:

- **Limpieza de datos:** Eliminar o corregir datos erróneos o incompletos.
- **Selección de características:** Elegir las variables más relevantes para el modelo.
- **Normalización o estandarización:** Transformar los datos para que tengan una escala común, lo cual es especialmente importante para algunos algoritmos que son sensibles a la escala de los datos, como la regresión lineal o los algoritmos basados en distancia.

•

2. División del Conjunto de Datos

Los datos generalmente se dividen en dos o tres conjuntos:

- **Conjunto de entrenamiento:** Utilizado para entrenar el modelo.
- **Conjunto de validación:** Utilizado para ajustar los parámetros del modelo y evitar el sobreajuste.
- **Conjunto de prueba:** Utilizado para evaluar la eficacia del modelo después del entrenamiento y la validación. No se utiliza durante el entrenamiento.

•

3. Elección del Modelo

Seleccionar el algoritmo adecuado para el problema que se quiere resolver. Esto depende de la naturaleza de los datos y del problema específico (por ejemplo, clasificación, regresión, clustering, etc.).

4. Entrenamiento del Modelo

Durante el entrenamiento, el algoritmo aprende a hacer predicciones o clasificaciones basándose en los datos. Esto se hace ajustando los parámetros del modelo para minimizar un error o maximizar una métrica de rendimiento a través de métodos como el descenso del gradiente.

5. Validación y Ajuste de Parámetros

Es común ajustar y afinar los parámetros del modelo, como la tasa de aprendizaje o el número de iteraciones. La validación cruzada es una técnica útil aquí para asegurar que el modelo no solo funciona bien en el conjunto de entrenamiento sino también en datos no vistos.

6. Evaluación del Modelo

Finalmente, el modelo se evalúa con el conjunto de prueba para ver cómo se desempeña con nuevos datos. Las métricas de evaluación pueden incluir precisión, recall, F1-score, entre otras, dependiendo del tipo de problema.

Cada uno de estos pasos es fundamental para asegurar que el modelo de machine learning sea robusto, eficiente y efectivo en la tarea para la que fue diseñado.

#11 Evaluación del modelo

¿Qué son las métricas de rendimiento de un modelo de Machine Learning?

Las métricas de rendimiento de un modelo de machine learning son herramientas estadísticas que se utilizan para evaluar y medir la eficacia de un modelo en realizar las tareas para las que fue entrenado, como la predicción o la clasificación. Estas métricas son esenciales para entender qué tan bien el modelo está trabajando con datos reales y para comparar diferentes modelos o configuraciones del mismo modelo. Las métricas de rendimiento varían dependiendo del tipo de problema (clasificación, regresión, clustering, etc.) y del contexto específico del modelo. Aquí te explico algunas de las más comunes:

Para Problemas de Clasificación

1. **Precisión (Accuracy):** Mide el porcentaje de predicciones correctas realizadas por el modelo sobre el total de casos evaluados.
2. **Precisión (Precision):** De todas las clasificaciones positivas que el modelo ha hecho, cuántas fueron realmente positivas.
3. **Recall (Sensibilidad):** De todos los casos positivos reales, cuántos fueron identificados correctamente por el modelo.
4. **F1-Score:** Es la media armónica de precisión y recall, útil cuando se necesita un balance entre estas dos métricas.
5. **AUC-ROC (Área bajo la Curva del Operador Receptor):** Mide la capacidad del modelo para distinguir entre clases. Un área de 1 representa una predicción perfecta, mientras que un área de 0.5 representa una predicción aleatoria.
- 6.

Para Problemas de Regresión

1. **Error Cuadrático Medio (MSE - Mean Squared Error):** Mide el promedio de los cuadrados de los errores, es decir, la diferencia cuadrada entre los valores observados y los valores predichos.
2. **Error Absoluto Medio (MAE - Mean Absolute Error):** Es el promedio de los valores absolutos de los errores.
3. **R cuadrado (R^2):** Proporciona una indicación de la bondad del ajuste y por lo tanto una medida de cuán bien se esperan que las muestras no vistas sean predichas por el modelo, a través de la proporción de la variación total de los resultados explicada por el modelo.
- 4.

Para Problemas de Clustering

1. **Coeficiente de Silueta:** Mide qué tan similar es un objeto a su propio cluster comparado con otros clusters.
2. **Índice Davies-Bouldin:** Evalúa la separación entre clusters, donde clusters más separados y compactos proporcionan mejores puntuaciones.
- 3.

Importancia de las Métricas

Las métricas permiten a los científicos de datos y desarrolladores:

- **Optimizar modelos:** Ajustando hiperparámetros basados en resultados de métricas específicas.
- **Control de calidad:** Verificando que el modelo cumple con los estándares de rendimiento requeridos para su despliegue.

- **Comparar modelos:** Decidiendo cuál modelo o configuración usar en función de su rendimiento en tareas específicas.

En resumen, las métricas de rendimiento son fundamentales para guiar el desarrollo de modelos de machine learning, ayudando a identificar los más efectivos y ajustarlos para maximizar su efectividad en la aplicación real.

Lo que sigue debajo de estas líneas es traducción de un capítulo de libro Luis G. Serrano “Grokking Machine Learning”

Tipos de aprendizaje automático (Capítulo 2)

- Tres diferentes tipos de aprendizaje automático: supervisado, no supervisado y por refuerzo.
- La diferencia entre datos etiquetados y no etiquetados.
- La diferencia entre regresión y clasificación, y cómo se utilizan.

El aprendizaje automático imita aproximadamente el proceso por el cual los humanos toman decisiones basadas en la experiencia, tomando decisiones basadas en datos anteriores. Naturalmente, programar computadoras para imitar el proceso de pensamiento humano es un desafío, porque las computadoras están diseñadas para almacenar y procesar números, no para tomar decisiones. Esta es la tarea que el aprendizaje automático pretende abordar. El aprendizaje automático se divide en varias ramas, dependiendo del tipo de decisión que se debe tomar. En este capítulo, revisamos algunas de las más importantes entre estas ramas.

El aprendizaje automático tiene aplicaciones en muchos campos, como los siguientes:

- Predecir precios de viviendas basado en el tamaño de la casa, número de habitaciones y ubicación.
- Predecir los precios del mercado de valores de hoy basado en los precios de ayer y otros factores del mercado.
- Detectar correos electrónicos spam y no spam basado en las palabras del correo electrónico y el remitente.
- Reconocer imágenes como caras o animales, basado en los píxeles de la imagen.
- Procesar documentos de texto largos y generar un resumen.
- Recomendar videos o películas a un usuario (por ejemplo, en YouTube o Netflix).
- Construir chatbots que interactúan con humanos y responden preguntas.
- Entrenar coches autónomos para navegar una ciudad por sí mismos.
- Diagnosticar pacientes como enfermos o sanos.
- Segmentar el mercado en grupos similares basados en ubicación, poder adquisitivo e intereses.
- Jugar juegos como el ajedrez o Go.

Intenta imaginar cómo podríamos usar el aprendizaje automático en cada uno de estos campos. Fíjate que algunas de estas aplicaciones son diferentes pero se pueden resolver de manera similar. Por ejemplo, predecir precios de viviendas y predecir precios de acciones se pueden hacer usando técnicas similares. Igualmente, predecir si un correo electrónico es spam y predecir si una transacción con tarjeta de crédito es legítima o fraudulenta también se pueden hacer usando técnicas similares. ¿Qué pasa con agrupar usuarios de una aplicación basado en su similitud? Eso suena diferente de predecir precios de viviendas, pero se podría hacer de manera similar a agrupar artículos de periódico por tema. ¿Y qué pasa con jugar al ajedrez? Eso suena diferente de todas las otras aplicaciones anteriores, pero podría ser como jugar al Go.

Los modelos de aprendizaje automático se agrupan en diferentes tipos, según la forma en que operan. Las tres familias principales de modelos de aprendizaje automático son:

- aprendizaje supervisado,
- aprendizaje no supervisado, y
- aprendizaje por refuerzo.

¿Cuál es la diferencia entre datos etiquetados y no etiquetados?

¿Qué son los datos?

Los datos son simplemente información. Cada vez que tenemos una tabla con información, tenemos datos. Normalmente, cada fila en nuestra tabla es un punto de datos. Digamos, por ejemplo, que tenemos un conjunto de datos de mascotas. En este caso, cada fila representa una mascota diferente. Cada mascota en la tabla está descrita por ciertas características de esa mascota.

¿Y qué son las características? Definimos las características como las propiedades o características de los datos. Si nuestros datos están en una tabla, las características son las columnas de la tabla. En nuestro ejemplo de mascotas, las características pueden ser el tamaño, el nombre, el tipo o el peso. Las características incluso podrían ser los colores de los píxeles en una imagen de la mascota. Esto es lo que describe nuestros datos.

Algunas características son especiales, sin embargo, y las llamamos etiquetas. ¿Etiquetas? Esto es un poco menos directo, porque depende del contexto del problema que estamos tratando de resolver.

Normalmente, si estamos tratando de predecir una característica particular basada en las otras, esa característica es la etiqueta. Si estamos tratando de predecir el tipo de mascota (por ejemplo, gato o perro) basado en la información sobre esa mascota, entonces la etiqueta es el tipo de mascota (gato/perro). Si estamos tratando de predecir si la mascota está enferma o sana basada en síntomas y otra información, entonces la etiqueta es el estado de la mascota (enfermo/sano). Si estamos tratando de predecir la edad de la mascota, entonces la etiqueta es la edad (un número).

Predicciones

Hemos estado usando el concepto de hacer predicciones libremente, pero ahora definámoslo claramente. El objetivo de un modelo de aprendizaje automático predictivo es adivinar las etiquetas en los datos.

La suposición que hace el modelo se llama predicción.

Ahora que sabemos qué son las etiquetas, podemos entender que hay dos tipos principales de datos: datos etiquetados y no etiquetados.

Datos etiquetados y no etiquetados

Los datos etiquetados son datos que vienen con etiquetas. Los datos no etiquetados son datos que no vienen con etiquetas. Un ejemplo de datos etiquetados es un conjunto de datos de correos electrónicos que viene con una columna que registra si los correos electrónicos son spam o no spam, o una columna que registra si el correo electrónico está relacionado con el trabajo. Un ejemplo de datos no etiquetados es un conjunto de datos de correos electrónicos que no tiene una columna particular en la que estemos interesados en predecir.

En la figura 2.1, vemos tres conjuntos de datos que contienen imágenes de mascotas. El primer conjunto de datos tiene una columna que registra el tipo de mascota, y el segundo conjunto de datos tiene una columna que especifica el peso de la mascota. Estos dos son ejemplos de datos etiquetados. El tercer conjunto de datos consiste solo en imágenes, sin etiquetas, lo que lo convierte en datos no etiquetados.

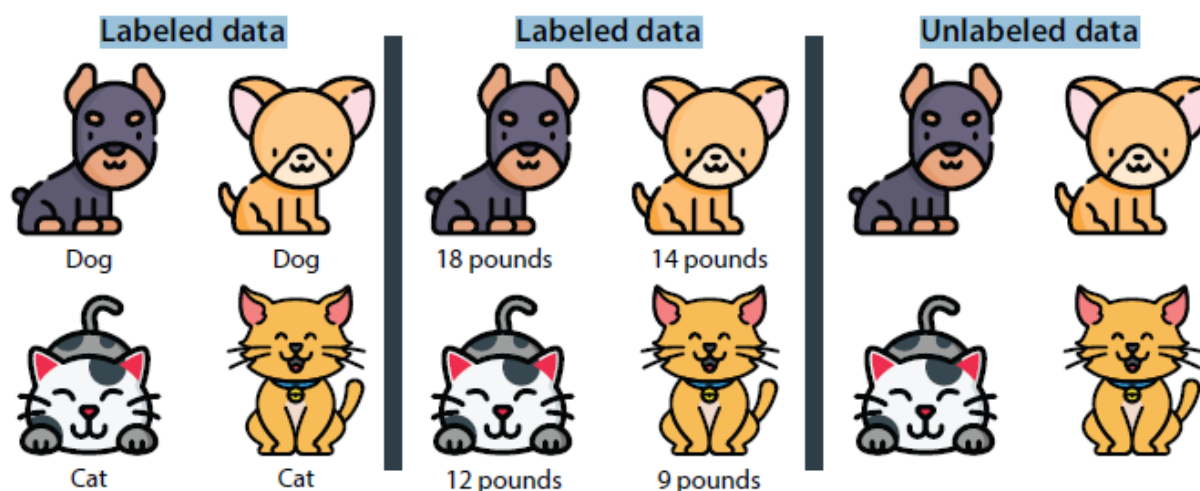


Figura 2.1: Los datos etiquetados son aquellos que vienen con una etiqueta o tag. Esa etiqueta puede ser un tipo o un número. Los datos no etiquetados son aquellos que no vienen con ninguna etiqueta. El conjunto de datos a la izquierda está etiquetado, y la etiqueta es el tipo de mascota (perro/gato). El conjunto de datos en el medio también está etiquetado, y la etiqueta es el peso de la mascota (en libras). El conjunto de datos a la derecha no está etiquetado.

Por supuesto, esta definición contiene cierta ambigüedad, porque dependiendo del problema, decidimos si una característica particular califica como una etiqueta. Por lo tanto, determinar si los datos están etiquetados o no, muchas veces, depende del problema que estamos tratando de resolver.

Los datos etiquetados y no etiquetados dan lugar a dos ramas diferentes del aprendizaje automático llamadas aprendizaje supervisado y aprendizaje no supervisado, que se definen en las siguientes tres secciones.

Aprendizaje supervisado: La rama del aprendizaje automático que trabaja con datos etiquetados.

Podemos encontrar aprendizaje supervisado en algunas de las aplicaciones más comunes hoy en día, incluyendo reconocimiento de imágenes, diversas formas de procesamiento de texto y sistemas de recomendación. El aprendizaje supervisado es un tipo de aprendizaje automático que utiliza datos etiquetados. En resumen, el objetivo de un modelo de aprendizaje supervisado es predecir (adivinar) las etiquetas.

En el ejemplo de la figura 2.1, el conjunto de datos de la izquierda contiene imágenes de perros y gatos, y las etiquetas son "perro" y "gato". Para este conjunto de datos, el modelo de aprendizaje automático utilizaría datos previos para predecir la etiqueta de nuevos puntos de datos. Esto significa que, si introducimos una nueva imagen sin etiqueta, el modelo adivinará si la imagen es de un perro o un gato, prediciendo así la etiqueta del punto de datos (figura 2.2).

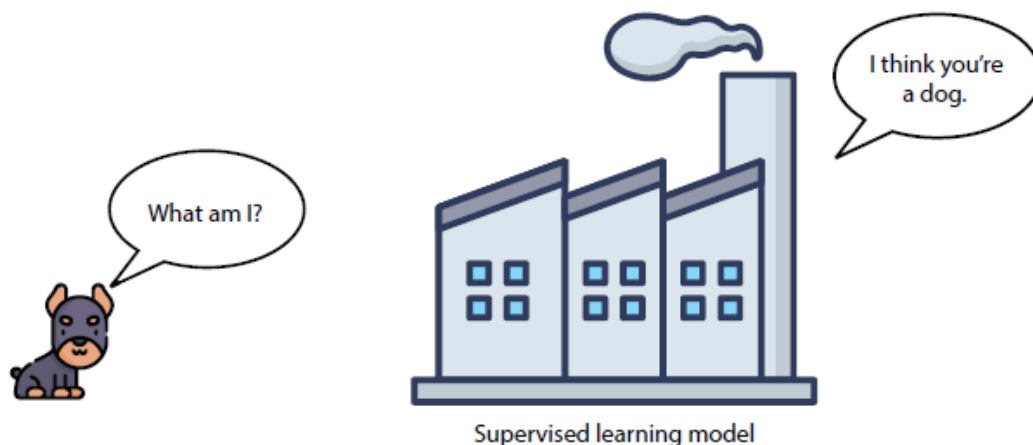


Figura 2.2 Un modelo de aprendizaje supervisado predice la etiqueta de un nuevo punto de datos. En este caso, el punto de datos corresponde a un perro, y el algoritmo de aprendizaje supervisado está entrenado para predecir que este punto de datos, efectivamente, corresponde a un perro.

Si recuerdas del capítulo 1, el marco que aprendimos para tomar una decisión fue recordar-formular-predicir. Esto es precisamente cómo funciona el aprendizaje supervisado. El modelo primero recuerda el conjunto de datos de perros y gatos. Luego formula un modelo, o una regla, sobre lo que cree que constituye un perro y un gato. Finalmente, cuando llega una nueva imagen, el modelo hace una predicción sobre lo que cree que es la etiqueta de la imagen, es decir, un perro o un gato (figura 2.3).

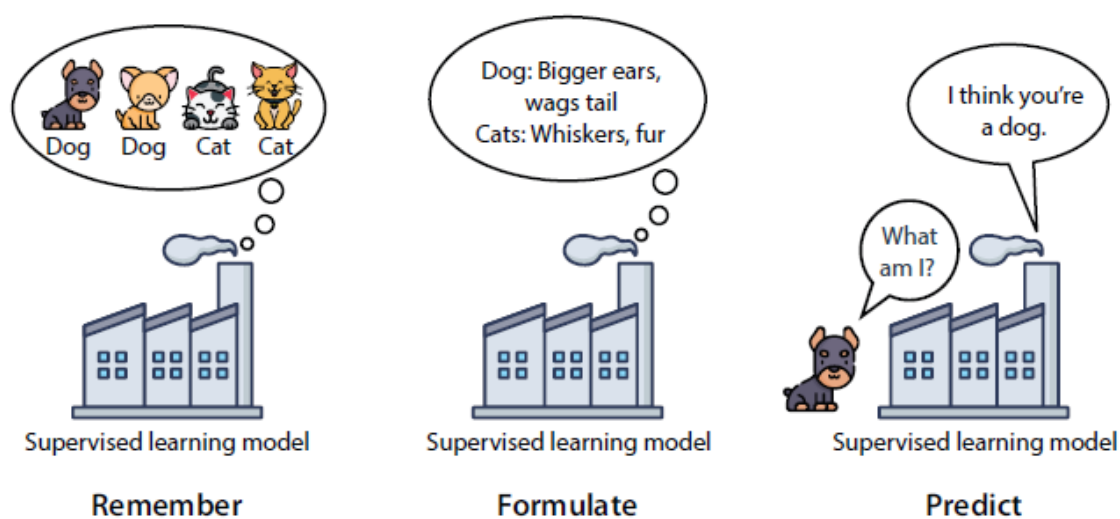


Figura 2.3 Un modelo de aprendizaje supervisado sigue el marco de recordar-formular-predecir del capítulo 1. Primero, recuerda el conjunto de datos. Luego, formula reglas sobre lo que constituiría un perro y un gato. Finalmente, predice si un nuevo punto de datos es un perro o un gato.

Ahora, observa que en la figura 2.1, tenemos dos tipos de conjuntos de datos etiquetados. En el conjunto de datos del medio, cada punto de datos está etiquetado con el peso del animal. En este conjunto de datos, las etiquetas son números. En el conjunto de datos de la izquierda, cada punto de datos está etiquetado con el tipo de animal (perro o gato). En este conjunto de datos, las etiquetas son estados. Los números y los estados son los dos tipos de datos que encontraremos en modelos de aprendizaje supervisado. Llamamos al primer tipo datos numéricos y al segundo tipo datos categóricos.

Los datos numéricos son cualquier tipo de datos que usan números, como 4, 2.35, o -199. Ejemplos de datos numéricos son precios, tamaños o pesos.

Los datos categóricos son cualquier tipo de datos que usan categorías, o estados, como masculino/femenino o gato/perro/pájaro. Para este tipo de datos, tenemos un conjunto finito de categorías para asociar a cada uno de los puntos de datos.

Esto da lugar a los siguientes dos tipos de modelos de aprendizaje supervisado:

- Modelos de regresión son los tipos de modelos que predicen datos numéricos. La salida de un modelo de regresión es un número, como el peso del animal.
- Modelos de clasificación son los tipos de modelos que predicen datos categóricos. La salida de un modelo de clasificación es una categoría, o un estado, como el tipo de animal (gato o perro).

Veamos dos ejemplos de modelos de aprendizaje supervisado, uno de regresión y uno de clasificación.

- Modelo 1: modelo de precios de vivienda (regresión). En este modelo, cada punto de datos es una casa. La etiqueta de cada casa es su precio. Nuestro objetivo es que cuando una nueva casa (punto de datos) sale al mercado, nos gustaría predecir su etiqueta, es decir, su precio.
- Modelo 2: modelo de detección de spam en correos electrónicos (clasificación). En este modelo, cada punto de datos es un correo electrónico. La etiqueta de cada correo electrónico es spam o no spam. Nuestro objetivo es que cuando llegue un nuevo correo electrónico (punto de datos) a nuestra bandeja de entrada, nos gustaría predecir su etiqueta, es decir, si es spam o no spam.

Observa la diferencia entre los modelos 1 y 2:

- El modelo de precios de vivienda es un modelo que puede devolver un número de muchas posibilidades, como \$100, \$250,000, o \$3,125,672.33. Por lo tanto, es un modelo de regresión.
- El modelo de detección de spam, por otro lado, solo puede devolver dos cosas: spam o no spam. Por lo tanto, es un modelo de clasificación.

En las siguientes subsecciones, elaboramos más sobre regresión y clasificación.

Los modelos de regresión predicen números

Como discutimos anteriormente, los modelos de regresión son aquellos en los cuales la etiqueta que queremos predecir es un número. Este número se predice en base a las características. En el ejemplo de viviendas, las características pueden ser cualquier cosa que describa una casa, como el tamaño, el número de habitaciones, la distancia a la escuela más cercana o la tasa de criminalidad en el barrio.

Otros lugares donde se pueden usar modelos de regresión incluyen:

- Mercado de valores: predecir el precio de una acción específica basado en otros precios de acciones y otros señales del mercado.
- Medicina: predecir la esperanza de vida esperada de un paciente o el tiempo de recuperación esperado, basado en los síntomas y el historial médico del paciente.
- Ventas: predecir la cantidad de dinero esperada que un cliente gastará, basado en la demografía del cliente y el comportamiento de compra pasado.
- Recomendaciones de videos: predecir la cantidad de tiempo esperada que un usuario verá un video, basado en la demografía del usuario y otros videos que ha visto.

El método más común utilizado para la regresión es la regresión lineal, que utiliza funciones lineales (líneas u objetos similares) para hacer nuestras predicciones basadas en las características.

Otros métodos populares utilizados para la regresión son la regresión de árboles de decisión, y varios métodos de ensamble como los bosques aleatorios, AdaBoost, árboles impulsados por gradiente y XGBoost.

Los modelos de clasificación predicen un estado

Los modelos de clasificación son aquellos en los que la etiqueta que queremos predecir es un estado que pertenece a un conjunto finito de estados. Los modelos de clasificación más comunes predicen un "sí" o un "no", pero muchos otros modelos utilizan un conjunto más amplio de estados. El ejemplo que vimos en la figura 2.3 es un ejemplo de clasificación, porque predice el tipo de la mascota, es decir, "gato" o "perro".

En el ejemplo de reconocimiento de spam en correos electrónicos, el modelo predice el estado del correo electrónico (es decir, spam o no spam) a partir de las características del correo electrónico. En este caso, las características del correo electrónico pueden ser las palabras que contiene, el número de errores ortográficos, el remitente o cualquier otra cosa que describa el correo electrónico.

Otra aplicación común de la clasificación es el reconocimiento de imágenes. Los modelos de reconocimiento de imágenes más populares toman como entrada los píxeles de la imagen y producen una predicción de lo que la imagen representa. Dos de los conjuntos de datos más famosos para el reconocimiento de imágenes son MNIST y CIFAR-10. MNIST contiene aproximadamente 60,000 imágenes en blanco y negro de 28 por 28

píxeles de dígitos escritos a mano que están etiquetados del 0 al 9. Estas imágenes provienen de una combinación de fuentes, incluyendo la Oficina del Censo de Estados Unidos y un repositorio de dígitos escritos a mano por estudiantes de secundaria estadounidenses. El conjunto de datos MNIST se puede encontrar en el siguiente enlace: <http://yann.lecun.com/exdb/mnist/>. El conjunto de datos CIFAR-10 contiene 60,000 imágenes en color de 32 por 32 píxeles de diferentes cosas. Estas imágenes están etiquetadas con 10 objetos diferentes (de ahí el 10 en su nombre), a saber, aviones, autos, pájaros, gatos, ciervos, perros, ranas, caballos, barcos y camiones. Esta base de datos es mantenida por el Instituto Canadiense de Investigación Avanzada (CIFAR), y se puede encontrar en el siguiente enlace: <https://www.cs.toronto.edu/~kriz/cifar.html>.

Algunas aplicaciones adicionales y poderosas de los modelos de clasificación incluyen:

- Análisis de sentimientos: predecir si una reseña de película es positiva o negativa, basado en las palabras de la reseña.
- Tráfico web: predecir si un usuario hará clic en un enlace o no, basado en la demografía del usuario y la interacción pasada con el sitio.
- Redes sociales: predecir si un usuario se hará amigo o interactuará con otro usuario, basado en su demografía, historia y amigos en común.
- Recomendaciones de videos: predecir si un usuario verá un video, basado en la demografía del usuario y otros videos que ha visto.

Aprendizaje no supervisado: La rama del aprendizaje automático que trabaja con datos no etiquetados.

El aprendizaje no supervisado también es un tipo común de aprendizaje automático. Se diferencia del aprendizaje supervisado en que los datos no están etiquetados. En otras palabras, el objetivo de un modelo de aprendizaje automático es extraer la mayor cantidad de información posible de un conjunto de datos que no tiene etiquetas o objetivos para predecir.

¿Qué podría ser tal conjunto de datos y qué podríamos hacer con él? En principio, podemos hacer un poco menos de lo que podemos hacer con un conjunto de datos etiquetado, porque no tenemos etiquetas para predecir. Sin embargo, aún podemos extraer mucha información de un conjunto de datos no etiquetado. Por ejemplo, volvamos al ejemplo de gatos y perros en el conjunto de datos más a la derecha en la figura 2.1. Este conjunto de datos consiste en imágenes de gatos y perros, pero no tiene etiquetas. Por lo tanto, no sabemos qué tipo de mascota representa cada imagen, así que no podemos predecir si una nueva imagen corresponde a un perro o un gato. Sin embargo, podemos hacer otras cosas, como determinar si dos imágenes son similares o diferentes. Esto es algo que hacen los algoritmos de aprendizaje no supervisado. Un algoritmo de aprendizaje no supervisado puede agrupar las imágenes basándose en la similitud, incluso sin saber qué representa cada grupo (figura 2.4). Si se hace correctamente, el algoritmo podría separar las imágenes de perros de las imágenes de gatos, ¡o incluso agrupar cada una de ellas por raza!

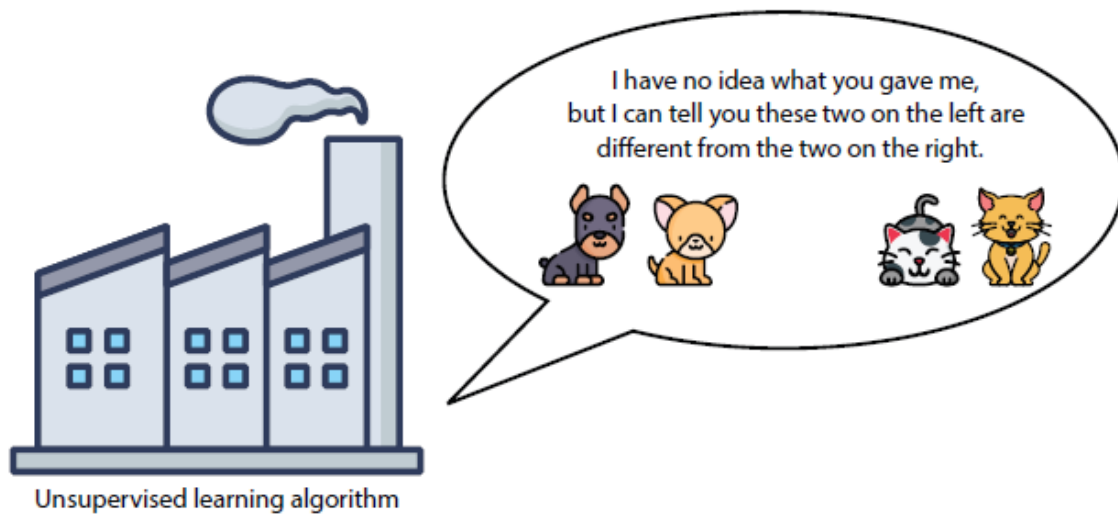


Figura 2.4 Un algoritmo de aprendizaje no supervisado aún puede extraer información de los datos. Por ejemplo, puede agrupar elementos similares.

De hecho, incluso si las etiquetas están presentes, aún podemos usar técnicas de aprendizaje no supervisado en nuestros datos para preprocesarlos y aplicar métodos de aprendizaje supervisado de manera más efectiva.

Las principales ramas del aprendizaje no supervisado son el agrupamiento, la reducción de dimensionalidad y el aprendizaje generativo.

- **Algoritmos de clustering (agrupamiento)**:

Son los algoritmos que agrupan los datos en clusters basados en la similitud.

- **Algoritmos de reducción de dimensionalidad**:

Son los algoritmos que simplifican nuestros datos y los describen fielmente con menos características.

- **Algoritmos generativos**:

Son los algoritmos que pueden generar nuevos puntos de datos que se asemejan a los datos existentes.

En las siguientes tres subsecciones, estudiamos estas tres ramas con más detalle.

Los algoritmos de agrupamiento dividen un conjunto de datos en grupos similares. Como mencionamos anteriormente, los algoritmos de agrupamiento son aquellos que dividen el conjunto de datos en grupos similares. Para ilustrar esto, volvamos a los dos conjuntos de datos en la sección "Aprendizaje supervisado": el conjunto de datos de viviendas y el conjunto de datos de correos electrónicos de spam, pero imaginemos que no tienen etiquetas. Esto significa que el conjunto de datos de viviendas no tiene precios, y el conjunto de datos de correos electrónicos no tiene información sobre si los correos son spam o no spam.

Comencemos con el conjunto de datos de viviendas. ¿Qué podemos hacer con este conjunto de datos? Aquí hay una idea: podríamos agrupar las casas por similitud. Por ejemplo, podríamos agruparlas por ubicación, precio, tamaño o una combinación de estos factores. Este proceso se llama agrupamiento. El agrupamiento es una rama del aprendizaje automático no supervisado que consiste en las tareas que agrupan los elementos en nuestro conjunto de datos en clústeres donde todos los puntos de datos son similares.

Ahora veamos el segundo ejemplo, el conjunto de datos de correos electrónicos. Dado que el conjunto de datos no está etiquetado, no sabemos si cada correo electrónico es spam o no spam. Sin embargo, todavía podemos aplicar algún agrupamiento a nuestro conjunto de datos. Un algoritmo de agrupamiento divide nuestras imágenes en algunos grupos diferentes basados en diferentes características del correo electrónico. Estas características podrían ser las palabras en el mensaje, el remitente, el número y tamaño de los adjuntos, o los tipos de enlaces dentro del correo electrónico. Después de agrupar el conjunto de datos, un humano (o una combinación de un humano y un algoritmo de aprendizaje supervisado) podría etiquetar estos clústeres por categorías como "Personal", "Social" y "Promociones".

Como ejemplo, veamos el conjunto de datos en la tabla 2.1, que contiene nueve correos electrónicos que nos gustaría agrupar. Las características del conjunto de datos son el tamaño del correo electrónico y el número de destinatarios.

Tabla 2.1 Una tabla de correos electrónicos con su tamaño y número de destinatarios.

Email	Size	Recipients
1	8	1
2	12	1
3	43	1
4	10	2
5	40	2
6	25	5
7	23	6
8	28	6
9	26	7

A simple vista, parece que podríamos agrupar los correos electrónicos por su número de destinatarios. Esto resultaría en dos clústeres: uno con correos electrónicos que tienen dos destinatarios o menos, y otro con correos electrónicos que tienen cinco o más destinatarios. También podríamos intentar agruparlos en tres grupos según el tamaño. Pero puedes imaginar que a medida que la tabla se hace más y más grande, distinguir los grupos se vuelve más y más difícil.

¿Y si graficamos los datos? Grafiquemos los correos electrónicos en un gráfico, donde el eje horizontal registre el tamaño y el eje vertical registre el número de destinatarios. Esto nos da el gráfico en la figura 2.5.

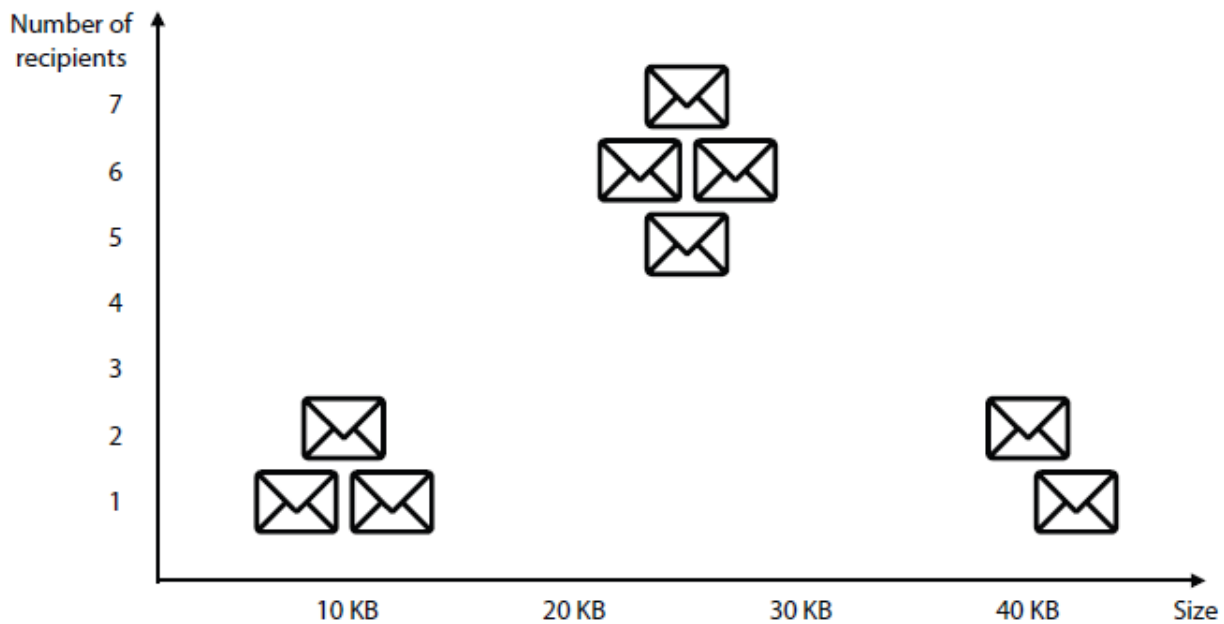


Figura 2.5: Un gráfico del conjunto de datos de correos electrónicos. El eje horizontal corresponde al tamaño del correo electrónico y el eje vertical al número de destinatarios. Podemos ver tres clústeres bien definidos en este conjunto de datos.

En la figura 2.5 podemos ver tres clústeres bien definidos, que están resaltados en la figura 2.6.

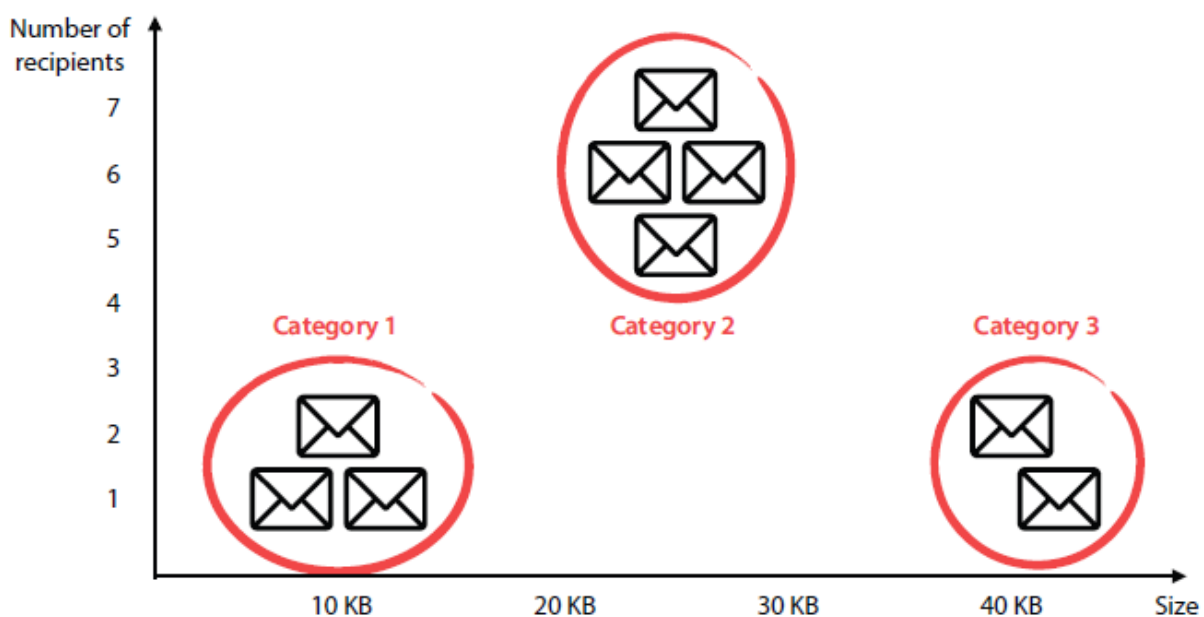


Figura 2.6: Podemos agrupar los correos electrónicos en tres categorías basadas en el tamaño y el número de destinatarios.

Más sobre modelos de aprendizaje no supervisado

En el resto de este libro, no cubrimos el aprendizaje no supervisado. Sin embargo, te animo encarecidamente a que lo estudies por tu cuenta. Aquí están algunos de los algoritmos de agrupamiento más importantes que

existen. El Apéndice C enumera varios más (incluyendo algunos videos míos) donde puedes aprender estos algoritmos en detalle.

- ****Agrupamiento K-medias****: este algoritmo agrupa puntos eligiendo algunos centros de masa aleatorios y moviéndolos cada vez más cerca de los puntos hasta que están en los lugares correctos.
- ****Agrupamiento jerárquico****: este algoritmo comienza agrupando los puntos más cercanos entre sí y continúa de esta manera, hasta que tenemos algunos grupos bien definidos.
- ****Agrupamiento espacial basado en densidad (DBSCAN)****: este algoritmo comienza agrupando puntos en lugares con alta densidad, mientras etiqueta los puntos aislados como ruido.
- ****Modelos de mezclas gaussianas****: este algoritmo no asigna un punto a un solo clúster, sino que asigna fracciones del punto a cada uno de los clústeres existentes. Por ejemplo, si hay tres clústeres, A, B y C, entonces el algoritmo podría determinar que el 60% de un punto particular pertenece al grupo A, el 25% al grupo B y el 15% al grupo C.

La reducción de dimensionalidad simplifica los datos sin perder demasiada información.

La reducción de dimensionalidad es un paso de preprocesamiento útil que podemos aplicar para simplificar enormemente nuestros datos antes de aplicar otras técnicas. Como ejemplo, volvamos al conjunto de datos de viviendas. Imagina que las características son las siguientes:

- Tamaño
- Número de dormitorios
- Número de baños
- Tasa de criminalidad en el vecindario
- Distancia a la escuela más cercana

Este conjunto de datos tiene cinco columnas de datos. ¿Qué pasaría si quisiéramos convertir el conjunto de datos en uno más simple con menos columnas, sin perder mucha información? Hagamos esto usando el sentido común. Observa más de cerca las cinco características. ¿Puedes ver alguna forma de simplificarlas, quizás agrupándolas en algunas categorías más pequeñas y más generales?

Después de observar detenidamente, podemos ver que las primeras tres características son similares, porque todas están relacionadas con el tamaño de la casa. De manera similar, la cuarta y quinta características son similares entre sí, porque están relacionadas con la calidad del vecindario. Podríamos condensar las primeras tres características en una gran característica de "tamaño", y la cuarta y quinta en una gran característica de "calidad del vecindario".

¿Cómo condensamos las características de tamaño? Podríamos olvidarnos de las habitaciones y dormitorios y considerar solo el tamaño, podríamos sumar el número de dormitorios y baños, o tal vez tomar alguna otra

combinación de las tres características. También podríamos condensar las características de calidad del área de maneras similares. Los algoritmos de reducción de dimensionalidad encontrarán buenas formas de condensar estas características, perdiendo la menor cantidad de información posible y manteniendo nuestros datos lo más intactos posible mientras logran simplificarlos para un proceso y almacenamiento más fáciles (figura 2.7).

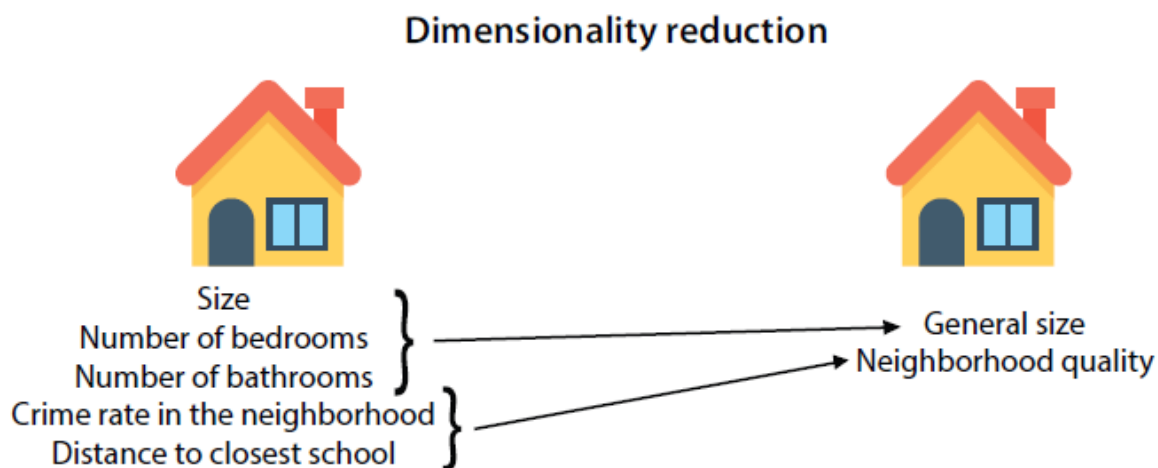


Figura 2.7 Los algoritmos de reducción de dimensionalidad nos ayudan a simplificar nuestros datos. A la izquierda, tenemos un conjunto de datos de viviendas con muchas características. Podemos usar la reducción de dimensionalidad para reducir el número de características en el conjunto de datos sin perder mucha información y obtener el conjunto de datos a la derecha.

¿Por qué se llama reducción de dimensionalidad si todo lo que estamos haciendo es reducir el número de columnas en nuestros datos? La palabra elegante para el número de columnas en un conjunto de datos es *dimensión*. Piensa en esto: si nuestros datos tienen una columna, entonces cada punto de datos es un número. Una colección de números se puede trazar como una colección de puntos en una línea, que tiene precisamente una dimensión.

Si nuestros datos tienen dos columnas, entonces cada punto de datos está formado por un par de números. Podemos imaginar una colección de pares de números como una colección de puntos en una ciudad, donde el primer número es el número de calle y el segundo número es la avenida. Las direcciones en un mapa son bidimensionales, porque están en un plano. ¿Qué pasa cuando nuestros datos tienen tres columnas? En este caso, cada punto de datos está formado por tres números. Podemos imaginar que si cada dirección en nuestra ciudad es un edificio, entonces los primeros dos números son la calle y la avenida, y el tercer número es el piso en el edificio. Esto parece más una ciudad tridimensional. Podemos seguir adelante

. ¿Qué pasa con cuatro números? Bueno, ahora realmente no podemos visualizarlo, pero si pudiéramos, este conjunto de puntos se vería como lugares en una ciudad cuatridimensional, y así sucesivamente. La mejor manera de imaginar una ciudad cuatridimensional es imaginando una tabla con cuatro columnas. ¿Qué pasa con una ciudad de 100 dimensiones? Esto sería una tabla con 100 columnas, en la que cada persona tiene una dirección que consiste en 100 números.

La imagen mental que podríamos tener al pensar en dimensiones superiores se muestra en la figura 2.8. Por lo tanto, al pasar de cinco dimensiones a dos, redujimos nuestra ciudad de cinco dimensiones a una ciudad bidimensional. Por eso se llama *reducción de dimensionalidad*.

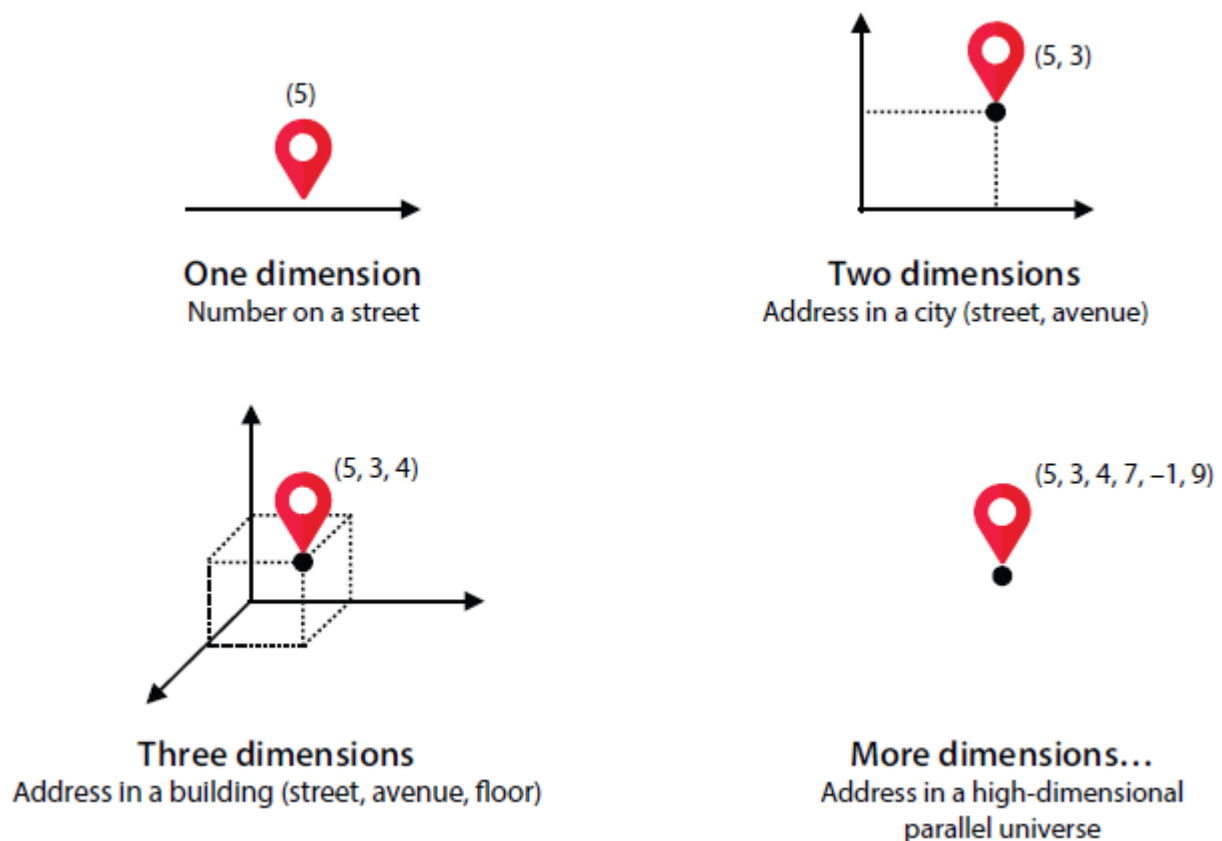


Figura 2.8 Cómo imaginar espacios de dimensiones superiores: Una dimensión es como una calle, en la que cada casa solo tiene un número. Dos dimensiones son como una ciudad plana, en la que cada dirección tiene dos números, una calle y una avenida. Tres dimensiones son como una ciudad con edificios, en la que cada dirección tiene tres números: una calle, una avenida y un piso. Cuatro dimensiones son como un lugar imaginario en el que cada dirección tiene cuatro números. Podemos imaginar dimensiones superiores como otra ciudad imaginaria en la que las direcciones tienen tantas coordenadas como necesitemos.

Otras formas de simplificar nuestros datos: Factorización de matrices y descomposición en valores singulares

Parece que el agrupamiento y la reducción de dimensionalidad no se parecen en nada, pero, en realidad, no son tan diferentes. Si tenemos una tabla llena de datos, cada fila corresponde a un punto de datos y cada columna corresponde a una característica. Por lo tanto, podemos usar el agrupamiento para reducir el número de filas en nuestro conjunto de datos y la reducción de dimensionalidad para reducir el número de columnas, como ilustran las figuras 2.9 y 2.10.

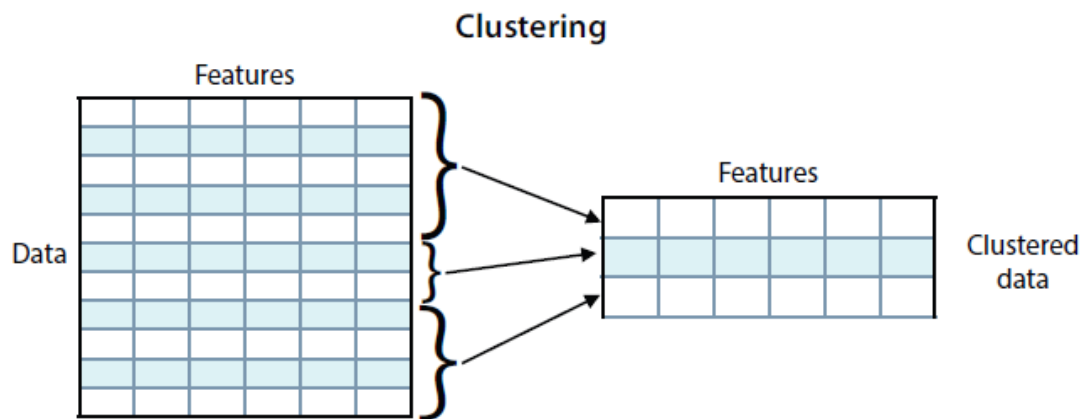


Figura 2.9: El agrupamiento se puede utilizar para simplificar nuestros datos reduciendo el número de filas en nuestro conjunto de datos al agrupar varias filas en una.

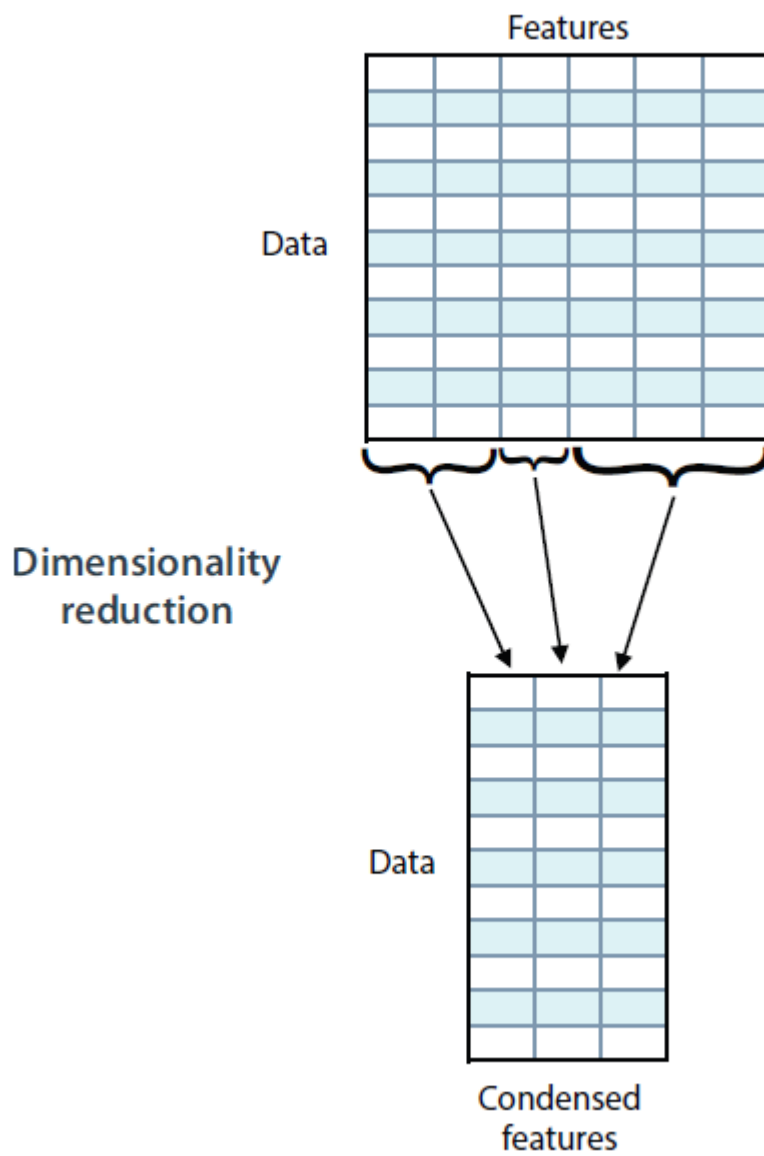


Figura 2.10: La reducción de dimensionalidad se puede utilizar para simplificar nuestros datos reduciendo el número de columnas en nuestro conjunto de datos.

Puedes estar preguntándote, ¿hay alguna manera de reducir tanto las filas como las columnas al mismo tiempo? ¡Y la respuesta es sí! Dos formas comunes de hacer esto son la factorización de matrices y la descomposición en valores singulares. Estos dos algoritmos expresan una gran matriz de datos en un producto de matrices más pequeñas.

Lugares como Netflix utilizan extensamente la factorización de matrices para generar recomendaciones. Imagina una gran tabla donde cada fila corresponde a un usuario, cada columna a una película, y cada entrada en la matriz es la calificación que el usuario dio a la película. Con la factorización de matrices, se pueden extraer ciertas características, como el tipo de película, los actores que aparecen en la película, y otros, y ser capaz de predecir la calificación que un usuario da a una película, basado en estas características.

La descomposición en valores singulares se utiliza en la compresión de imágenes. Por ejemplo, una imagen en blanco y negro se puede ver como una gran tabla de datos, donde cada entrada contiene la intensidad del píxel correspondiente. La descomposición en valores singulares utiliza técnicas de álgebra lineal para simplificar esta tabla de datos, lo que nos permite simplificar la imagen y almacenar su versión más simple utilizando menos entradas.

Aprendizaje automático generativo

El aprendizaje automático generativo es uno de los campos más asombrosos del aprendizaje automático. Si has visto caras ultra-realistas, imágenes o videos creados por computadoras, entonces has visto el aprendizaje automático generativo en acción.

El campo del aprendizaje generativo consiste en modelos que, dados unos datos, pueden generar nuevos puntos de datos que parecen muestras de ese conjunto de datos original. Estos algoritmos están obligados a aprender cómo se ven los datos para producir puntos de datos similares. Por ejemplo, si el conjunto de datos contiene imágenes de rostros, entonces el algoritmo producirá rostros de aspecto realista. Los algoritmos generativos han logrado crear imágenes extremadamente realistas, pinturas, y mucho más. También han generado video, música, historias, poesía y muchas otras cosas maravillosas. El algoritmo generativo más popular son las redes generativas antagónicas (GANs, por sus siglas en inglés), desarrolladas por Ian Goodfellow y sus coautores. Otros algoritmos generativos útiles y populares son los autoencoders variacionales, desarrollados por Kingma y Welling, y las máquinas de Boltzmann restringidas (RBMs, por sus siglas en inglés), desarrolladas por Geoffrey Hinton.

Como puedes imaginar, el aprendizaje generativo es bastante difícil. Para un humano, es mucho más fácil determinar si una imagen muestra un perro que dibujar uno. Esta tarea es igual de difícil para las computadoras. Por lo tanto, los algoritmos en el aprendizaje generativo son complicados, y se necesita mucha capacidad de cálculo y datos para hacer que funcionen bien. Dado que este libro se centra en el aprendizaje supervisado, no cubriremos el aprendizaje generativo en detalle, pero en el capítulo 10, obtendremos una idea de cómo funcionan algunos de estos algoritmos generativos, porque tienden a usar redes neuronales.

¿Qué es el aprendizaje por refuerzo?

El aprendizaje por refuerzo es un tipo diferente de aprendizaje automático en el que no se proporcionan datos y debemos lograr que la computadora realice una tarea. En lugar de datos, el modelo recibe un entorno y un agente que se supone debe navegar en este entorno. El agente tiene un objetivo o un conjunto de objetivos. El entorno cuenta con recompensas y castigos que guían al agente para tomar las decisiones correctas y alcanzar su objetivo. Todo esto puede sonar un poco abstracto, pero veamos un ejemplo.

Ejemplo: Mundo de rejilla (Grid World)

En la figura 2.11, vemos un mundo de rejilla con un robot en la esquina inferior izquierda. Ese es nuestro agente. El objetivo es llegar al cofre del tesoro en la esquina superior derecha de la rejilla. En la rejilla, también podemos ver una montaña, lo que significa que no podemos pasar por esa casilla, porque el robot no puede escalar montañas. También vemos un dragón, que atacará al robot, si el robot se atreve a aterrizar en su casilla, lo que significa que parte de nuestro objetivo es no aterrizar allí. Este es el juego. Y para darle al robot información sobre cómo proceder, llevamos un registro de la puntuación. La puntuación comienza en cero. Si el robot llega al cofre del tesoro, entonces ganamos 100 puntos. Si el robot alcanza al dragón, perdemos 50 puntos. Y para asegurarnos de que nuestro robot se mueva rápidamente, podemos decir que por cada paso que el robot da, perdemos 1 punto, porque el robot pierde energía al caminar.

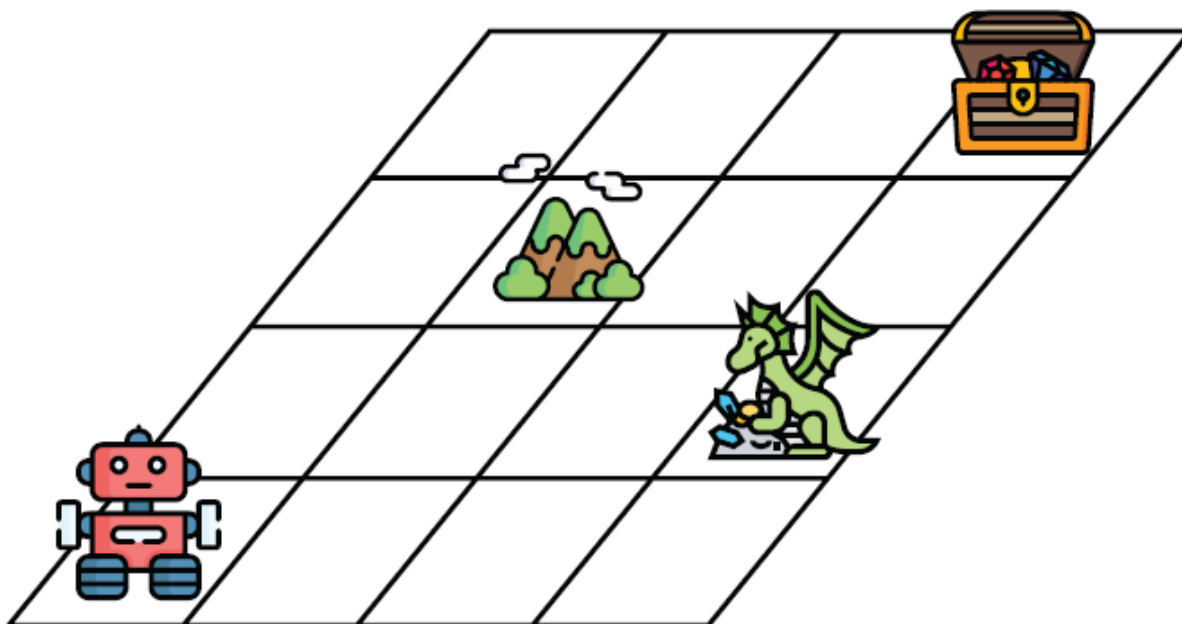


Figura 2.11: Un mundo de rejilla en el que nuestro agente es un robot. El objetivo del robot es encontrar el cofre del tesoro, mientras evita al dragón. La montaña representa un lugar por el cual el robot no puede pasar.

La forma de entrenar este algoritmo, en términos muy generales, es la siguiente: El robot comienza a caminar alrededor, registrando su puntuación y recordando qué pasos lo llevaron allí. Después de un tiempo, puede encontrarse con el dragón, perdiendo muchos puntos. Por lo tanto, aprende a asociar la casilla del dragón y las casillas cercanas a ella con puntuaciones bajas. En algún momento también puede alcanzar el cofre del tesoro, y aprende a comenzar a asociar esa casilla y las casillas cercanas a ella con puntuaciones altas. Después de jugar este juego durante mucho tiempo, el robot tendrá una buena idea de cuán buena es cada casilla, y puede tomar el camino siguiendo las casillas hasta el cofre del tesoro. La Figura 2.12 muestra un camino posible, aunque este no es ideal, porque pasa demasiado cerca del dragón. ¿Puedes pensar en uno mejor?

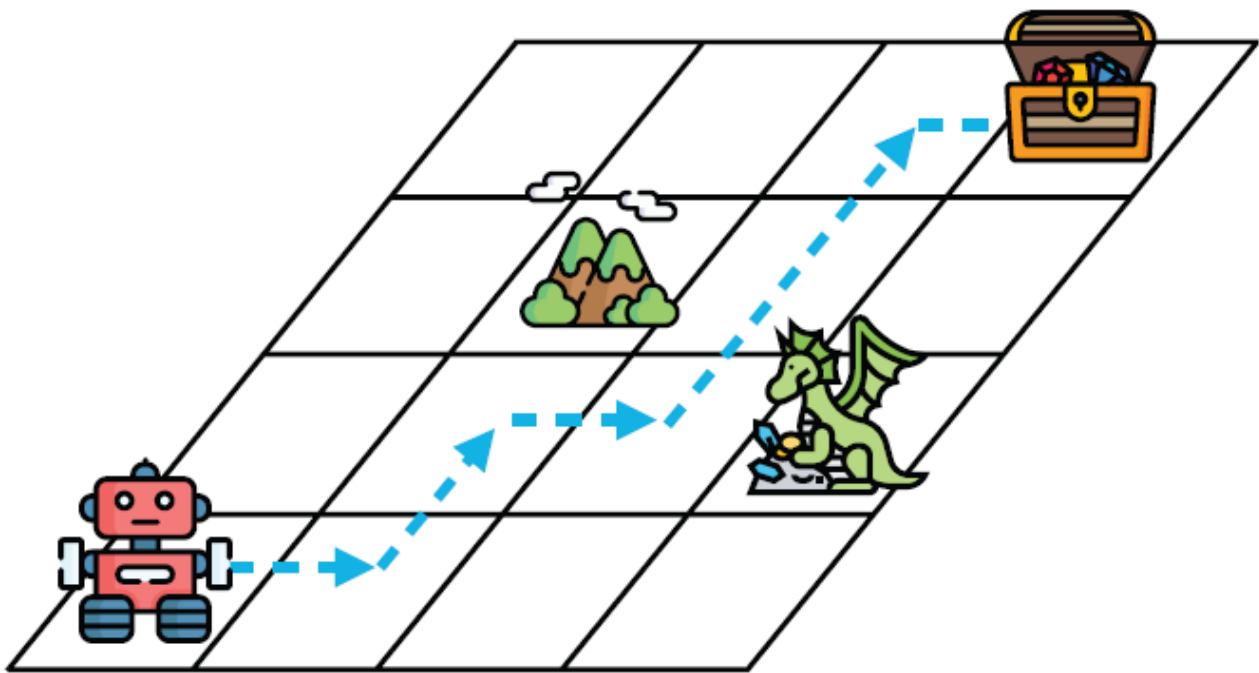


Figura 2.12: Aquí hay un camino que el robot podría tomar para encontrar el cofre del tesoro.

Por supuesto, esta es una explicación muy breve, y hay mucho más que aprender sobre el aprendizaje por refuerzo. El Apéndice C recomienda algunos recursos para estudios adicionales, incluyendo un video sobre aprendizaje por refuerzo profundo.

El aprendizaje por refuerzo tiene numerosas aplicaciones de vanguardia, incluyendo las siguientes:

- **Juegos**: avances recientes en enseñar a las computadoras cómo ganar en juegos, como Go o ajedrez, utilizan el aprendizaje por refuerzo. También se ha enseñado a agentes a ganar en juegos de Atari como Breakout o Super Mario.
- **Robótica**: el aprendizaje por refuerzo se utiliza extensamente para ayudar a los robots a llevar a cabo tareas como levantar cajas, limpiar una habitación o incluso bailar.
- **Coches autónomos**: las técnicas de aprendizaje por refuerzo se utilizan para ayudar al coche a llevar a cabo muchas tareas, como la planificación de rutas o el comportamiento en entornos particulares.

Dibujando una línea cerca de nuestros puntos: Regresión lineal (capítulo 3)

La regresión lineal es un método poderoso y ampliamente utilizado para estimar valores, como el precio de una casa, el valor de una acción determinada, la esperanza de vida de una persona, o la cantidad de tiempo que un usuario verá un video o pasará en un sitio web. Puede que ya hayas visto la regresión lineal antes como una plétora de fórmulas complicadas que incluyen derivadas, sistemas de ecuaciones y determinantes. Sin embargo, también podemos ver la regresión lineal de una manera más gráfica y menos formulística. En este capítulo, para entender la regresión lineal, todo lo que necesitas es la capacidad de visualizar puntos y líneas moviéndose.

Digamos que tenemos algunos puntos que más o menos parecen estar formando una línea, como se muestra en la figura 3.1.

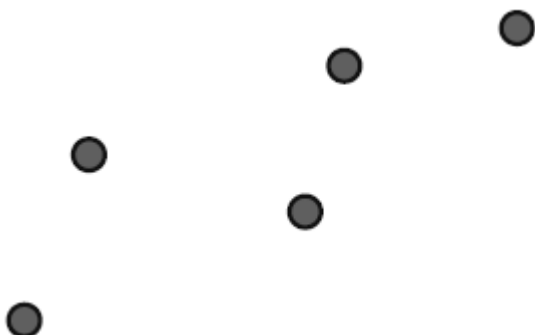


Figura 3.1: Algunos puntos que más o menos parecen estar formando una línea.

El objetivo de la regresión lineal es trazar la línea que pase lo más cerca posible de estos puntos. ¿Qué línea trazaría que pasara cerca de esos puntos? ¿Qué te parece la que se muestra en la figura 3.2?

Imagina que los puntos son casas en un pueblo, y nuestro objetivo es construir una carretera que atravesase el pueblo. Queremos que la línea pase lo más cerca posible de los puntos porque los habitantes del pueblo quieren vivir cerca de la carretera, y nuestro objetivo es complacerlos tanto como podamos.

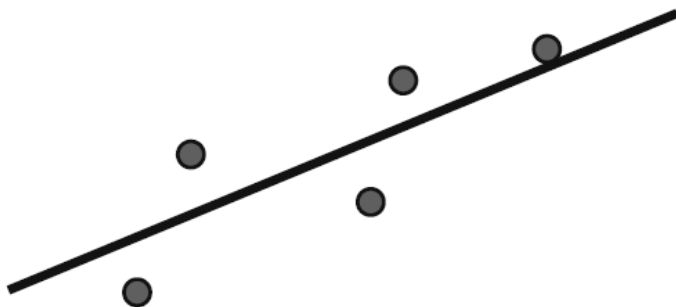


Figura 3.2: Una línea que pasa cerca de los puntos.

También podemos imaginar los puntos como imanes fijados al suelo (por lo que no pueden moverse). Ahora imagina lanzar una varilla de metal recta sobre ellos. La varilla se moverá, pero debido a que los imanes la atraen, eventualmente terminará en una posición de equilibrio, tan cerca como pueda de todos los puntos.

Por supuesto, esto puede llevar a mucha ambigüedad. ¿Queremos una carretera que pase algo cerca de todas las casas, o tal vez realmente cerca de algunas de ellas y un poco más lejos de otras? Algunas preguntas que surgen son:

- ¿Qué queremos decir con "puntos que más o menos parecen estar formando una línea"?
- ¿Qué queremos decir con "una línea que pasa realmente cerca de los puntos"?
- ¿Cómo encontramos tal línea?
- ¿Por qué esto es útil en el mundo real?
- ¿Por qué esto es aprendizaje automático?

En este capítulo respondemos todas estas preguntas y construimos un modelo de regresión lineal para predecir los precios de las viviendas en un conjunto de datos real.

El problema: Necesitamos predecir el precio de una casa

Digamos que somos agentes inmobiliarios encargados de vender una nueva casa. No conocemos el precio y queremos inferirlo comparándolo con otras casas. Observamos características de la casa que podrían influir en el precio, como el tamaño, el número de habitaciones, la ubicación, la tasa de criminalidad, la calidad de las escuelas y la distancia al comercio. Al final del día, queremos una fórmula para todas estas características que nos dé el precio de la casa, o al menos una buena estimación de él.

La solución: Construir un modelo de regresión para precios de viviendas

Vamos con un ejemplo lo más simple posible. Observamos solo una de las características: el número de habitaciones. Nuestra casa tiene cuatro habitaciones, y hay seis casas cercanas, con una, dos, tres, cinco, seis y siete habitaciones, respectivamente. Sus precios se muestran en la tabla 3.1.

Tabla 3.1 Una tabla de casas con el número de habitaciones y precios. La Casa 4 es aquella cuyo precio estamos tratando de inferir.

Number of rooms	Price
1	150
2	200
3	250
4	?
5	350
6	400
7	450

¿Qué precio le darías a la casa 4, basándote únicamente en la información de esta tabla? Si dijiste \$300, entonces hicimos la misma suposición. Probablemente viste un patrón y lo usaste para inferir el precio de la casa. Lo que hiciste en tu cabeza fue regresión lineal. Estudiemos más este patrón. Puede que hayas notado que cada vez que añades una habitación, se añaden \$50 al precio de la casa. Más específicamente, podemos pensar en el precio de una casa como una combinación de dos cosas: un precio base de \$100 y un cargo extra de \$50 por cada una de las habitaciones. Esto se puede resumir en una fórmula simple:

$$\text{Precio} = 100 + 50 \times (\text{Número de habitaciones})$$

Lo que hicimos aquí es crear un modelo representado por una fórmula que nos da una predicción del precio de la casa, basada en la característica, que es el número de habitaciones. El precio por habitación se llama el peso de esa característica correspondiente, y el precio base se llama el sesgo del modelo. Estos son todos conceptos importantes en el aprendizaje automático. Aprendimos algunos de ellos en los capítulos 1 y 2, pero refresquemos nuestra memoria definiendo estos términos desde la perspectiva de este problema.

****Características (features)**:**

Las características de un punto de datos son aquellas propiedades que utilizamos para hacer nuestra predicción. En este caso, las características son el número de habitaciones en la casa, la tasa de criminalidad, la edad de la casa, el tamaño, y así sucesivamente. Para nuestro caso, hemos decidido por una característica: el número de habitaciones en la casa.

****Etiquetas (labels)**:**

Esta es el objetivo que intentamos predecir a partir de las características. En este caso, la etiqueta es el precio de la casa.

****Modelo**:**

Un modelo de aprendizaje automático es una regla o una fórmula que predice una etiqueta a partir de las características. En este caso, el modelo es la ecuación que encontramos para el precio.

****Predicción**:**

La predicción es la salida del modelo. Si el modelo dice, "Creo que la casa con cuatro habitaciones va a costar \$300", entonces la predicción es 300.

****Pesos (weights)**:**

En la fórmula correspondiente al modelo, cada característica se multiplica por un factor correspondiente. Estos factores son los pesos. En la fórmula anterior, la única característica es el número de habitaciones, y su peso correspondiente es 50.

****Sesgo (bias)**:**

Como puedes ver, la fórmula correspondiente al modelo tiene una constante que no está adjunta a ninguna de las características. Esta constante se llama sesgo. En este modelo, el sesgo es 100 y corresponde al precio base de una casa.

Ahora, la pregunta es, ¿cómo llegamos a esta fórmula? O más específicamente, ¿cómo hacemos que la computadora llegue a este peso y sesgo? Para ilustrar esto, veamos un ejemplo un poco más complicado. Y dado que este es un problema de aprendizaje automático, lo abordaremos usando el marco recordar-formular-predecir que aprendimos en el capítulo 2. Más específicamente, recordaremos los precios de otras casas, formularemos un modelo para el precio y usaremos este modelo para predecir el precio de una nueva casa.

****El paso de recordar**:**

Mirando los precios de las casas existentes

Para ver el proceso más claramente, echemos un vistazo a un conjunto de datos un poco más complicado, como el de la tabla 3.2.

Tabla 3.2 Un conjunto de datos un poco más complicado de casas con su número de habitaciones y su precio.

Number of rooms	Price
1	155
2	197
3	244
4	?
5	356
6	407
7	448

Este conjunto de datos es similar al anterior, excepto que ahora los precios no siguen un patrón agradable, donde cada precio es \$50 más que el anterior. Sin embargo, no está tan lejos del conjunto de datos original, por lo que podemos esperar que un patrón similar debería aproximar bien estos valores.

Normalmente, lo primero que hacemos cuando obtenemos un nuevo conjunto de datos es graficarlo. En la figura 3.3, podemos ver un gráfico de los puntos en un sistema de coordenadas en el que el eje horizontal representa el número de habitaciones y el eje vertical representa el precio de la casa.

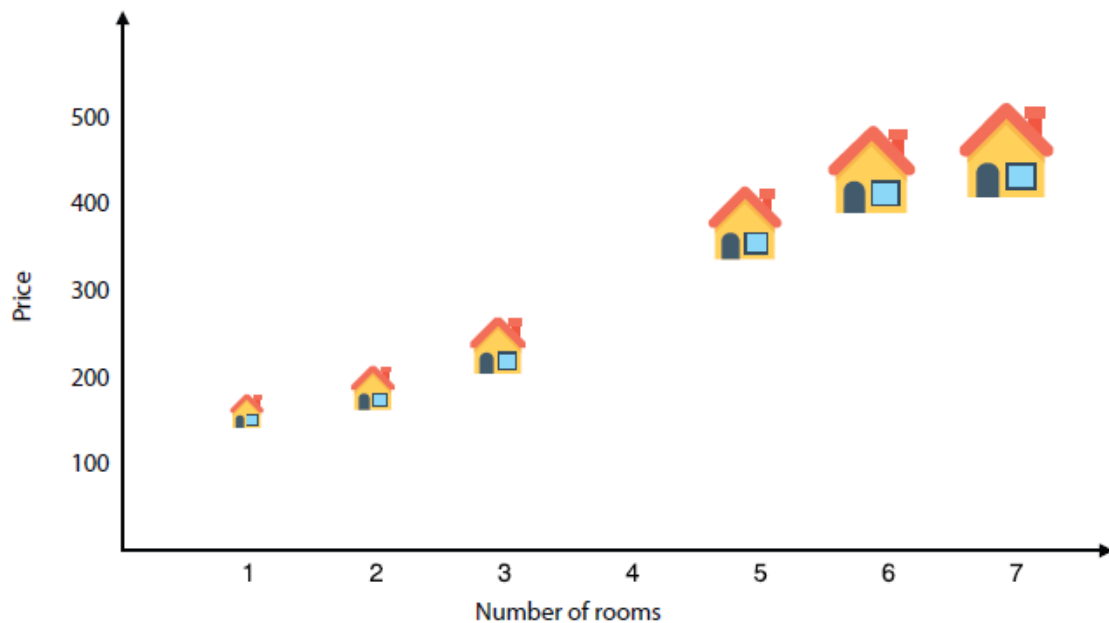


Figura 3.3: Gráfico del conjunto de datos en la tabla 3.2. El eje horizontal representa el número de habitaciones, y el eje vertical representa el precio de la casa.

****El paso de formular**:**

Formulando una regla que estime el precio de la casa

El conjunto de datos en la tabla 3.2 es lo suficientemente cercano al de la tabla 3.1, por lo que por ahora, podemos sentirnos seguros usando la misma fórmula para el precio. La única diferencia es que ahora los precios no son exactamente lo que la fórmula dice, y tenemos un pequeño error. Podemos escribir la ecuación de la siguiente manera:

$$\text{Precio} = 100 + 50(\text{Número de habitaciones}) + (\text{Error pequeño})$$

Si queremos predecir precios, podemos usar esta ecuación. Aunque no estamos seguros de obtener el valor real, sabemos que es probable que nos acerquemos. Ahora la pregunta es, ¿cómo encontramos esta ecuación? Y lo más importante, ¿cómo encuentra esta ecuación una computadora?

Volvamos al gráfico y veamos qué significa la ecuación allí. ¿Qué sucede si miramos todos los puntos en los que la coordenada vertical (y) es 100 más 50 veces la coordenada horizontal (x)? Este conjunto de puntos forma una línea con pendiente 50 e intersección y de 100. Antes de desglosar la declaración anterior, aquí están las definiciones de pendiente, intersección 'y' y la ecuación de una línea. Nos adentramos en estos en más detalle en la sección "Curso intensivo sobre pendiente e intersección y".

****Pendiente**:**

La pendiente de una línea es una medida de cuán empinada es. Se calcula dividiendo el aumento sobre la carrera (es decir, cuántas unidades sube, dividido por cuántas unidades va hacia la derecha). Esta proporción

es constante a lo largo de toda la línea. En un modelo de aprendizaje automático, esto es el peso de la característica correspondiente, y nos dice cuánto esperamos que suba el valor de la etiqueta, cuando aumentamos el valor de la característica en una unidad. Si la línea es horizontal, entonces la pendiente es cero, y si la línea desciende, la pendiente es negativa.

****Intersección con el eje Y**:**

La intersección con el eje Y de una línea es la altura a la que la línea cruza el eje vertical (Y). En un modelo de aprendizaje automático, es el sesgo y nos dice qué sería la etiqueta en un punto de datos donde todas las características son exactamente cero.

****Ecuación lineal**:**

Esta es la ecuación de una línea. Se da por dos parámetros: la pendiente y la intersección con el eje Y. Si la pendiente es m y la intersección con el eje Y es b , entonces la ecuación de la línea es $y = mx + b$, y la línea está formada por todos los puntos (x, y) que satisfacen la ecuación. En un modelo de aprendizaje automático, x es el valor de la característica y y es la predicción para la etiqueta. El peso y el sesgo del modelo son m y b , respectivamente.

Ahora podemos analizar la ecuación. Cuando decimos que la pendiente de la línea es 50, esto significa que cada vez que agregamos una habitación a la casa, estimamos que el precio de la casa aumentará en \$50. Cuando decimos que la intersección con el eje Y de la línea es 100, esto significa que la estimación para el precio de una casa (hipotética) con cero habitaciones sería el precio base de \$100. Esta línea está dibujada en la figura 3.4.



Figura 3.4: El modelo que formulamos es la línea que pasa lo más cerca posible de todas las casas.

Ahora, de todas las posibles líneas (cada una con su propia ecuación), ¿por qué elegimos esta en particular?

Porque pasa cerca de los puntos. Puede haber una mejor, pero al menos sabemos que esta es buena, a diferencia de una que no pasa cerca de los puntos en absoluto. Ahora volvemos al problema original, donde tenemos un conjunto de casas, y queremos construir una carretera lo más cerca posible de ellas.

¿Cómo encontramos esta línea? Lo veremos más adelante en el capítulo. Pero por ahora, digamos que tenemos una bola de cristal que, dada un montón de puntos, encuentra la línea que pasa más cerca de ellos.

El paso de predicción: ¿Qué hacemos cuando una nueva casa sale al mercado?

Ahora, vamos a usar nuestro modelo para predecir el precio de la casa de cuatro habitaciones. Para esto, introducimos el número cuatro como característica en nuestra fórmula para obtener lo siguiente:

$$\text{Precio} = 100 + 50 \cdot 4 = 300$$

Por lo tanto, nuestro modelo predice que la casa cuesta \$300. Esto también se puede ver gráficamente usando la línea, como se ilustra en la figura 3.5.

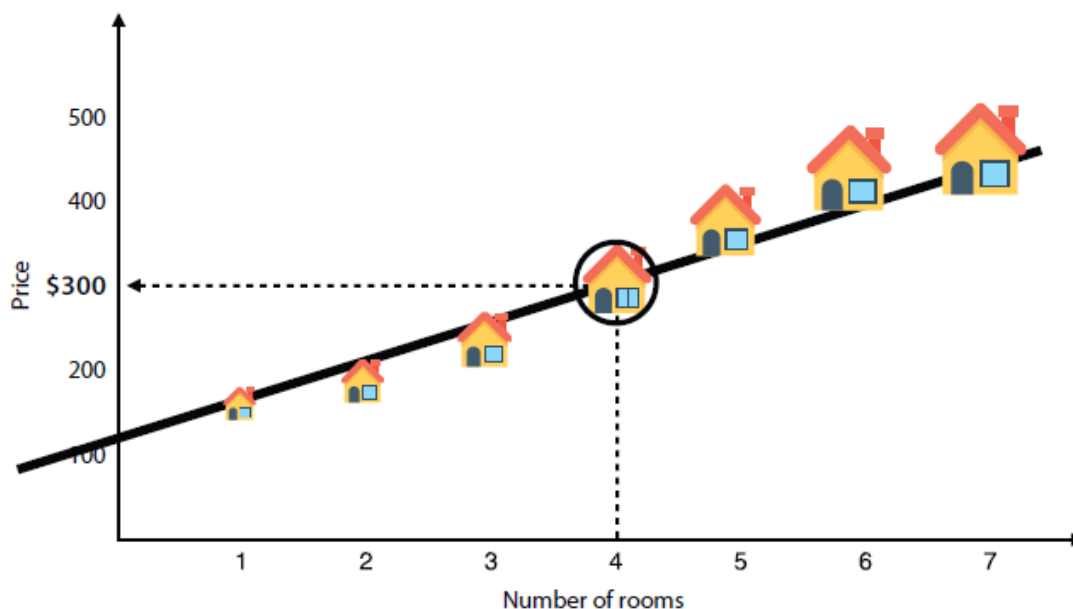


Figura 3.5: Nuestra tarea ahora es predecir el precio de la casa con cuatro habitaciones. Usando el modelo (línea), deducimos que el precio predicho de esta casa es de \$300.

¿Qué pasa si tenemos más variables? Regresión lineal multivariable

En las secciones anteriores aprendimos sobre un modelo que predice el precio de una casa basado en una característica: el número de habitaciones. Podemos imaginar muchas otras características que podrían ayudarnos a predecir el precio de una casa, como el tamaño, la calidad de las escuelas en el vecindario y la edad de la casa. ¿Puede nuestro modelo de regresión lineal acomodar estas otras variables? Absolutamente. Cuando la única característica es el número de habitaciones, nuestro modelo predice el precio como la suma de la característica multiplicada por su peso correspondiente, más un sesgo. Si tenemos más características,

todo lo que necesitamos hacer es multiplicarlas por sus pesos correspondientes y añadirlas al precio predicho. Por lo tanto, un modelo para el precio de una casa podría verse así:

$$\text{Precio} = 30(\text{número de habitaciones}) + 1.5(\text{tamaño}) + 10(\text{calidad de las escuelas}) - 2(\text{edad de la casa}) + 50$$

En esta ecuación, ¿por qué todos los pesos son positivos, excepto el que corresponde a la edad de la casa? La razón es que las otras tres características (número de habitaciones, tamaño y calidad de las escuelas) están correlacionadas positivamente con el precio de la casa. En otras palabras, debido a que las casas que son más grandes y están bien ubicadas cuestan más, cuanto mayor sea esta característica, mayor esperamos que sea el precio de la casa. Sin embargo, porque podríamos imaginar que las casas más antiguas tienden a ser menos costosas, la característica de la edad está correlacionada negativamente con el precio de la casa.

¿Qué pasa si el peso de una característica es cero? Esto sucede cuando una característica es irrelevante para el precio. Por ejemplo, imagina una característica que mide el número de vecinos cuyo apellido comienza con la letra A. Esta característica es mayormente irrelevante para el precio de la casa, por lo que esperaríamos que en un modelo razonable, el peso correspondiente a esta característica sea cero o algo muy cercano a ello.

De manera similar, si una característica tiene un peso muy alto (ya sea negativo o positivo), interpretamos esto como que el modelo nos indica que esa característica es importante para determinar el precio de la casa. En el modelo anterior, parece que el número de habitaciones es una característica importante, porque su peso es el más grande (en valor absoluto).

En la sección llamada “La reducción de dimensionalidad simplifica los datos sin perder demasiada información” en el capítulo 2, relacionamos el número de columnas en un conjunto de datos con la dimensión en la que vive el conjunto de datos. Así, un conjunto de datos con dos columnas se puede representar como un conjunto de puntos en el plano, y un conjunto de datos con tres columnas se puede representar como un conjunto de puntos en el espacio tridimensional. En tal conjunto de datos, un modelo de regresión lineal no corresponde a una línea sino a un plano que pasa lo más cerca posible de los puntos. Imagina tener muchas moscas volando alrededor en la habitación en una posición estacionaria, y nuestra tarea es tratar de pasar una gigantesca hoja de cartón lo más cerca que podamos de todas las moscas. Esto es regresión lineal multivariable con tres variables. El problema se vuelve difícil de visualizar para conjuntos de datos con más columnas, pero siempre podemos imaginar una ecuación lineal con muchas variables.

En este capítulo, principalmente tratamos con el entrenamiento de modelos de regresión lineal con solo una característica, pero el procedimiento es similar con más características. Te animo a leer sobre esto mientras mantienes este hecho en mente, e imagina cómo generalizarías cada una de nuestras próximas afirmaciones a un caso con varias características.

CONTINUARA....

