Resumen del capítulo: Recopilación de datos

Fuentes de datos

Hay muchas fuentes de datos que puedes usar para entrenar modelos.

Una de las fuentes es el almacén de datos de la empresa. Todo lo que tenemos que hacer es extraer ese historial y prepararnos para el análisis. Pero ese es un escenario bastante optimista.

La mayoría de las veces, la empresa no puede proporcionar los datos. Si la tarea es común, los conjuntos de datos se pueden encontrar en fuentes abiertas:

- Plataforma de competencia Kaggle Data Science;
- Repositorio de machine learning UC Irvine;
- Base de datos abierta del gobierno de los EE. UU.;
- FiveThirtyEight: datos abiertos sobre análisis de encuestas de opinión, política, economía y más.

Para algunas tareas, los datos se pueden recopilar en Internet. Descarga todas las páginas del portal requerido y usa el software **crawler** o **scraper** para extraer los datos.

Etiquetado de datos

Quizás la empresa tiene los datos, pero le faltan los valores objetivo. Dichos datos **no están etiquetados**, pero sí puedes usarlos. Para obtener un conjunto de entrenamiento, debemos realizar **etiquetado de datos** o **anotación de datos**. Es decir, establecer la respuesta correcta para cada foto (una persona o algo más). Esto **etiquetará** efectivamente los datos. En ocasiones, el etiquetado se puede realizar sin conocimientos especiales (como es el caso de las fotos de perfil), pero cuando los datos se relacionan con la salud y el bienestar de las personas, es posible que necesites asesoría profesional.

Hay servicios en línea dedicados para el etiquetado. Los datos sin etiquetar se cargan en el recurso y se especifica el precio del etiquetado por observación. Cualquiera puede ir al servicio, etiquetar los datos y cobrar por ello.

Estos son algunos servicios populares:

- Amazon Mechanical Turk
- Y.Toloka

Control de calidad de etiquetado

La calidad de los datos después del etiquetado se puede mejorar utilizando los métodos para el **control de calidad del etiquetado**. ¿Cómo funcionan? Todas las observaciones, o una parte de ellas, se etiquetan varias veces y luego se forma la respuesta final.

Veamos uno de esos métodos, el **voto mayoritario**. ¿Quién está "votando" y cómo? Por ejemplo, cada observación está etiquetada por tres evaluadores. La respuesta final es la elegida por la mayoría.

Fuga de información

Ya recopilaste los datos. Ahora podemos verificar si hay **fugas de información**. Ocurre cuando la información sobre el objetivo se filtra accidentalmente en las funciones.

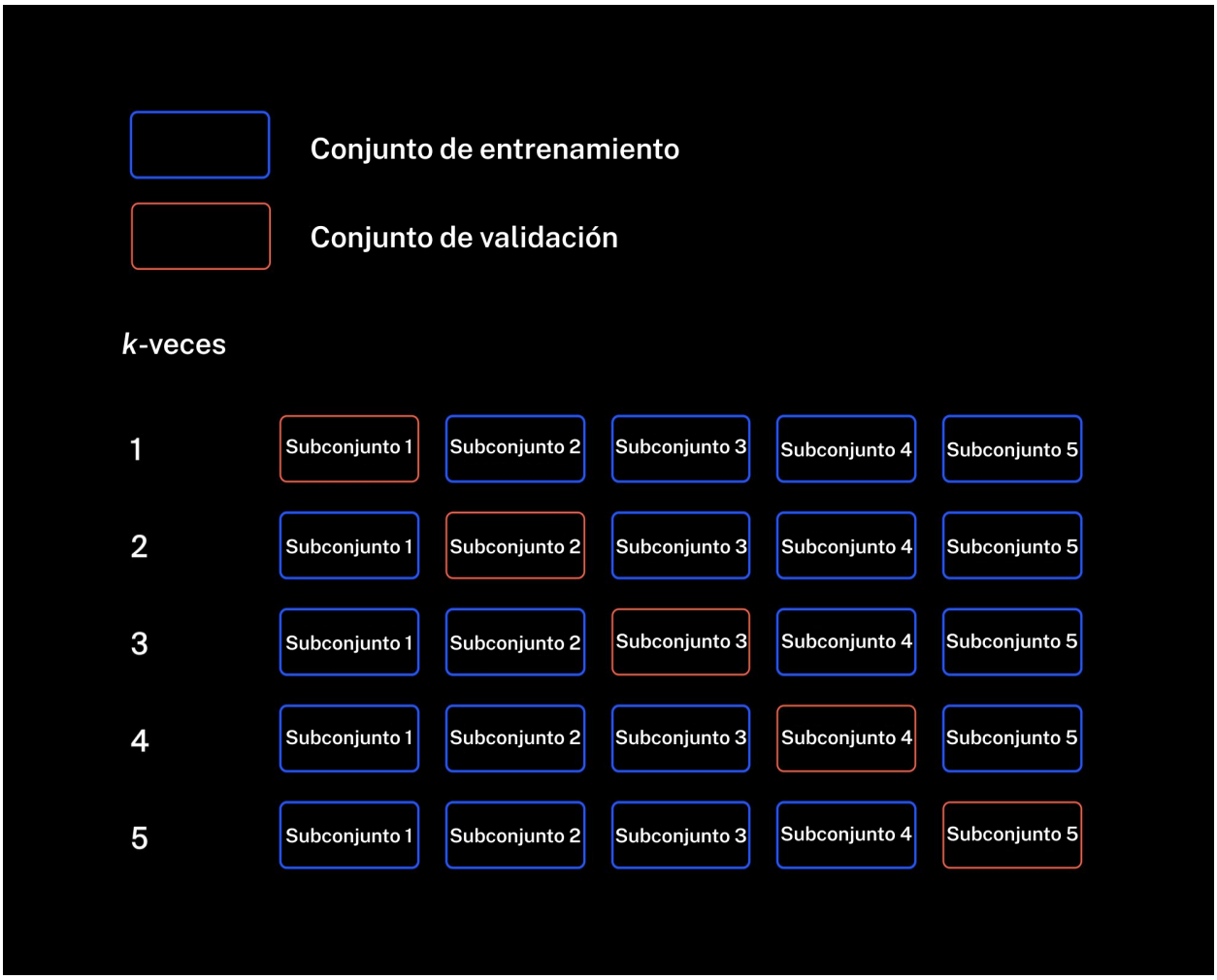
Validación cruzada

Ya sabes cómo formar muestras de entrenamiento, prueba y validación, y que la representatividad se logra a través del muestreo aleatorio. Pero ¿cómo podemos asegurarnos de que la distribución sea correcta en un conjunto de datos grande? ¡Toma varias muestras al azar!

La **validación cruzada** te ayudará a entrenar y probar el modelo utilizando varias muestras formadas al azar.

¿Como funciona? Dividimos todos los datos en conjunto de entrenamiento y conjunto de prueba. Mantenemos el conjunto de prueba hasta la evaluación final y dividimos aleatoriamente el conjunto de entrenamiento en *K* bloques iguales. El método de división en sí se llama *K-Fold*, donde *K* es el número de bloques o pliegues (de ahí el nombre).

Supongamos que en la primera etapa del procedimiento, el "Bloque 1" es el conjunto de validación y el resto de los bloques son para entrenamiento. En la segunda etapa, el "Bloque 2" se usa para la validación y el resto de los bloques se usan para el entrenamiento. Avanzando de esta manera, cada bloque llega a ser el conjunto de validación. Entonces el proceso se repite *K* veces.



Estamos "cruzando" los datos, tomando cada vez un nuevo bloque para la validación. Y la media de todos los valores obtenidos a través de la validación cruzada es el puntaje de evaluación de nuestro modelo.

El método de validación cruzada se asemeja a un bootstrap en el que se forman varias muestras, pero la diferencia es que la validación cruzada usa subconjuntos con contenido fijo que no cambia en cada etapa de entrenamiento y validación. Cada observación pasa por el conjunto de entrenamiento y el conjunto de validación.

La validación cruzada es útil cuando necesitamos comparar modelos, seleccionar hiperparámetros o evaluar la utilidad de las funciones. Minimiza la aleatoriedad de la división de datos y proporciona un resultado más preciso. El único inconveniente de la validación cruzada es el tiempo de cálculo, especialmente con muchas observaciones o un valor alto de *K*. Es mucho tiempo.

Validación cruzada en Sklearn

La validación cruzada puede llevar menos tiempo si usamos las herramientas de *sklearn*. Para evaluar el modelo por validación cruzada usaremos la función **cross_val_score** del módulo *sklearn.model_selection*.

Así se llama a la función:

```
from sklearn.model_selection import cross_val_score
cross_val_score(model, features, target, cv=3)
```

La función toma varios argumentos, como:

- *model*: modelo para validación cruzada. Está entrenado en el proceso de validación cruzada, por lo que tenemos que pasarlo sin entrenar. Supongamos que necesitamos este modelo para un árbol de decisión:

```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
```

- *features*
- *target*
- *cv* — número de bloques para validación cruzada (son 3, por defecto)

La función no requiere dividir los datos en bloques o muestras para la validación y el entrenamiento. Todos estos pasos se realizan de forma automática. La función devuelve una lista de valores de evaluación del modelo de cada validación. Cada valor es igual a *model.score()* para la muestra de validación. Por ejemplo, para una tarea de clasificación, esto es *exactitud*.