Resumen del capítulo: Ingeniería de características

Conocer los datos

Tanto las conversiones implícitas (de un tipo de datos derivado al tipo de datos padre: por ejemplo, de int a float, o 5 → 5.0) como las conversiones explícitas (lo contrario) son importantes. Es esencial que los datos se presenten en un formato conveniente y legible.

Una parte de este proceso es nombrar columnas. Para esto sirve el método `set_axis()`.

Clasificar por tipo

Las categorías en un conjunto de datos podrían estar almacenadas como strings de diversas longitudes.

¿Cuáles son las implicaciones de almacenarlas de este modo?

- La tabla es difícil de procesar visualmente.
- El tamaño del archivo y el tiempo de procesamiento de datos es mayor de lo necesario.
- Para filtrar datos por tipo de ticket, necesitamos ingresar el nombre completo (¡sin ningún error de dedo!).
- Crear nuevas categorías y modificar las existentes puede tomar demasiado tiempo.

Para almacenar información sobre categorías de la mejor manera posible, utiliza un diccionario que mapee cada nombre de categoría como un número. Este número se utilizará en la tabla en lugar del nombre de la categoría.

Clasificar por grupo de edad

Suele haber solamente una entrada con un valor de índice específico. Es imposible trabajar con bits de datos como este y llegar a conclusiones estadísticas. Es por ello que estos datos deben **categorizarse**, es decir, organizarse en categorías.

Una manera de categorizar los datos es filtrarlos por grupo de edad. Por ejemplo, 18 o menos, 19-65, o más de 65.

Reglas de clasificación como estas pueden representarse de manera práctica en Python como funciones que toman parámetros y devuelven un valor de categoría.

La función `group` que escribimos y el método `apply()` pueden usarse para devolver una columna con un grupo basado en una columna con un índice distinto.

```
data['column_group'] = data['column'].apply(group)
```

Funciones de una fila

Cuando el valor de una columna individual no es suficiente para categorizar, la función puede pasar los contenidos de una fila completa como un objeto Series. Una función a la que se le da una fila completa también puede devolver un valor de una columna en específico.

Cuando se procesan filas en lugar de valores individuales, el método `apply()` se diferencia en dos aspectos:

1. Se llama al método `apply()` para el DataFrame `data`, no solo para la columna `['age']`.
2. Por defecto, pandas pasa las columnas a la función `group()`. Para pasar filas a una función, debemos usar el método `apply()` con el parámetro `axis = 1`.