One-Hot 編碼轉換解說

編撰：屏東大學 周國華老師 (與 Google Gemini 共筆) 2025/12/02

在準備訓練 XGBoost 模型的過程中，**One-Hot 編碼（One-Hot Encoding）** 是最關鍵的第一步。

這個方法專門用來處理我們資料集中的 **類別型 (Categorical)** 欄位，例如 Department（部門）和 Category（消費類別）。

1. 什麼是 One-Hot 編碼？

One-Hot 編碼是一種將**文字或類別資料**轉換為**數值格式**的方法，這是因為大多數機器學習演算法（包括 XGBoost）只能處理數值。

核心機制：

針對一個含有 K 種不同類別的欄位，One-Hot 編碼會將其轉換成 K 個新的**二元 (Binary)** 欄位。

- 每筆資料在轉換後的這 K 個欄位中，**只有一個欄位的值會是 1** (表示「有」這個類別)，其他欄位的值都是 0 (表示「無」)。

2. 為什麼需要 One-Hot 編碼？（以部門為例）

假設我們的 Department 欄位有三個類別：Sales、R&D、HR。

交易編號	原始欄位：**Department**
T1	Sales
T2	R&D
T3	HR

如果我們簡單地將其編碼為數值：Sales=1, R&D=2, HR=3，就會產生以下問題：

- **假定順序/大小關係 (Ordinality)**：模型會誤以為 3 > 2 > 1，認為 HR 部門比 Sales 部門「更大」或「更重要」，但在實務上，這些部門是平等、沒有順序關係的。

One-Hot 編碼解決了這個問題，將三個類別轉換成三個獨立、平等的特徵：

交易編號	**Department_Sales**	**Department_R&D**	**Department_HR**
T1	**1**	0	0
T2	0	**1**	0
T3	0	0	**1**

現在，每個類別都是一個獨立的特徵，模型可以公正地學習「只有 R&D 部門」的交易對舞弊機率的影響，避免了錯誤的大小關係假設。

3. 對 XGBoost 模型的意義

雖然像 XGBoost 這樣的**樹狀模型 (Tree-based Models)** 對於數值順序的魯棒性（Robustness）比線性模型（如 Logit）要高，但在處理多達數十種或數百種的類別特徵時，使用 One-Hot 編碼仍然是業界最標準且最穩健的做法。

它允許 XGBoost 在建樹時，對每一個類別特徵（例如 Department_Sales）單獨進行條件判斷（例如：如果 Department_Sales 等於 1，往左邊分枝；如果等於 0，往右邊分枝）。