資料探勘

711378912 蔡宜諠

04/10/2025

[學習紀錄]

1. 屬性轉換的原因
   -> 讓不同尺度或單位的資料可比較、提升模型穩定性。
2. 屬性轉換的方法
   -> 正規化（Normalization）、標準化（Standardization）、對數轉換等。
3. 正規化與標準化分別為何
   -> 正規化：縮放到固定範圍（如 0–1）。
   -> 標準化：轉成平均 0、標準差 1 的分佈。
4. 屬性萃取的原因
   -> 將多變數簡化為少數有代表性的特徵，減少維度。
5. 屬性萃取的方法
   -> 主成分分析（PCA）、線性判別分析（LDA）。
6. 屬性挑選的原因
   -> 去除無關或重複特徵，提高模型效率與準確度。
7. 屬性挑選的方法
   -> 逐步選取、LASSO、隨機森林重要度分析。

屬性轉換移除

```
# 設定隨機種子，讓結果可重現
set.seed(1234)

# 產生 3x4 隨機矩陣（3列4行）
X <- matrix(runif(12), 3)

# 檢查矩陣內容
X
```

```
##           [,1]      [,2]        [,3]      [,4]
## [1,] 0.1137034 0.6233794 0.009495756 0.5142511
## [2,] 0.6222994 0.8609154 0.232550506 0.6935913
## [3,] 0.6092747 0.6403106 0.666083758 0.5449748
```

```
# 橫向(每一列)加總
X_sum <- apply(X, 1, sum)
X_sum
```

```
## [1] 1.260830 2.409357 2.460644
```

```
# [1] 1.261 2.409 2.461

# 方法一：每一列除以自己的加總 → 正規化到「列總和 = 1」
X_norm1 <- X / X_sum
X_norm1
```

```
##            [,1]      [,2]        [,3]      [,4]
## [1,] 0.09018142 0.4944200 0.007531355 0.4078672
## [2,] 0.25828448 0.3573217 0.096519754 0.2878741
## [3,] 0.24760784 0.2602207 0.270694898 0.2214765
```

```
# 方法二：換算成百分比（每列總和 = 100）
X_norm100 <- X / X_sum * 100
X_norm100
```

```
##           [,1]     [,2]       [,3]     [,4]
## [1,]  9.018142 49.44200  0.7531355 40.78672
## [2,] 25.828448 35.73217  9.6519754 28.78741
## [3,] 24.760784 26.02207 27.0694898 22.14765
```

```
# 方法三（補充）：z-score 標準化（每列均值為0，標準差為1）
X_z <- t(apply(X, 1, scale))
X_z
```

```
##             [,1]      [,2]       [,3]       [,4]
## [1,] -0.67336325 1.0298142 -1.0215925  0.6651416
## [2,]  0.07502749 0.9719476 -1.3899777  0.3430025
## [3,] -0.11268173 0.4814445  0.9748254 -1.3435882
```

```
# 設定隨機種子
set.seed(1234)

# 建立 3x4 隨機矩陣
X <- matrix(runif(12), 3)
X
```

```
##           [,1]      [,2]        [,3]      [,4]
## [1,] 0.1137034 0.6233794 0.009495756 0.5142511
## [2,] 0.6222994 0.8609154 0.232550506 0.6935913
## [3,] 0.6092747 0.6403106 0.666083758 0.5449748
```

```
# ✅ 方法一：橫向正規化（以最大值為1）
X_max <- apply(X, 1, max)   # 每列最大值
X_max
```

```
## [1] 0.6233794 0.8609154 0.6660838
```

```
# [1] 0.6234 0.8609 0.6661

# 每列除以最大值 → 最大變成 1
X_max_norm <- X / X_max

X_max100 <- X / X_max * 100
X_max100
```

```
##          [,1]      [,2]       [,3]     [,4]
## [1,] 18.23984 100.00000   1.523271 82.49408
## [2,] 72.28346 100.00000  27.012005 80.56440
## [3,] 91.47119  96.13064 100.000000 81.81776
```

```
apply(X_max100, 1, max)
```

```
## [1] 100 100 100
```

橫向正規化

```
# 設定隨機種子
set.seed(1234)

# 建立 3x4 隨機矩陣
X <- matrix(runif(12), 3)
X
```

```
##           [,1]      [,2]        [,3]      [,4]
## [1,] 0.1137034 0.6233794 0.009495756 0.5142511
## [2,] 0.6222994 0.8609154 0.232550506 0.6935913
## [3,] 0.6092747 0.6403106 0.666083758 0.5449748
```

```
# 計算每一列的向量長度（平方和再開根號）
X_length <- sqrt(apply(X^2, 1, sum))
X_length
```

```
## [1] 0.8161341 1.2897986 1.2336444
```

```
# 每一列除以其向量長度 → 使每一列的長度為 1
X_length1 <- X / X_length
X_length1
```

```
##           [,1]      [,2]       [,3]      [,4]
## [1,] 0.1393195 0.7638199 0.01163504 0.6301062
## [2,] 0.4824780 0.6674805 0.18029986 0.5377516
## [3,] 0.4938820 0.5190398 0.53993173 0.4417601
```

```
# 驗證：每一列平方後加總皆為 1
apply(X_length1^2, 1, sum)
```

```
## [1] 1 1 1
```