

# データサイエンス レポート課題

25G1026 大菅 海兎

## 選択した分析手法

回帰分析

## 例題

2024 年発売の PlayStation5 用フルプライスタイトル (8000 円から 10000 円) の Metacritic による評価と、大手中古販売会社のオンラインショップ価格を 100 分の 1 にし、四捨五入してまとめた。下記データについて中古販売価格と評価の回帰直線を求めよ。ただし、説明変数  $x$  を中古販売価格、目的変数  $y$  を Meta スコアとしなさい。

表 1 中古価格と評価データ

|          | A  | B  | C  | D  | E  | F  | G  | H  |
|----------|----|----|----|----|----|----|----|----|
| 販売価格     | 18 | 39 | 34 | 43 | 44 | 46 | 48 | 15 |
| Meta スコア | 82 | 87 | 94 | 92 | 86 | 94 | 81 | 79 |

## 解答

回帰分析は、最小二乗法から求められる標本回帰直線を利用する。この直線は、式 1 のように表される。

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x \quad (1)$$

まずは、各変数の平均値を求める。 $x$  の平均値  $\bar{x}$  は式 2 のように求められる。また、 $y$  の平均値  $\bar{y}$  は式 3 のように求められる。

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{287}{8} = 35.875 \quad (2)$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i = \frac{695}{8} = 86.875 \quad (3)$$

次に、回帰係数  $\hat{\beta}_1$  を式 4 のように求める。

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} = \frac{268.8750}{1134.8750} = 0.2369 \quad (4)$$

最後に、切片  $\hat{\beta}_0$  を式 5 のように求める。

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} = 86.875 - 0.2369 \times 35.875 = 78.4064 \quad (5)$$

よって、求める標本回帰直線は式 6 のようになる。

$$\hat{y} = 78.4064 + 0.2369x \quad (6)$$

式 6 から、中古販売価格が高いほど Meta スコアも高くなる傾向があるといえる。

---

#### ソースコード 1 プログラム例

---

```
1 import numpy as np
2
3 # 1. データの準備
4 x = np.array([18, 39, 34, 43, 44, 46, 48, 15]) # 価格
5 y = np.array([82, 87, 94, 92, 86, 94, 81, 79]) # 評価
6
7 # 2. 基本統計量の計算
8 n = len(x)
9 x_mean = np.mean(x)
10 y_mean = np.mean(y)
11
12 # 3. 偏差、偏差の二乗、偏差の積の計算
13 x_diff = x - x_mean # xの偏差
14 y_diff = y - y_mean # yの偏差
15 x_diff_sq = x_diff **2 # xの偏差平方
16 xy_diff = x_diff * y_diff # 偏差の積
17
18 # 4. 合計値の計算
19 sum_x_diff_sq = np.sum(x_diff_sq) # xの偏差平方和
20 sum_xy_diff = np.sum(xy_diff) # 偏差共分散
21
22 # 5. 回帰係数の算出
23 beta_1_hat = sum_xy_diff / sum_x_diff_sq
24 beta_0_hat = y_mean - (beta_1_hat * x_mean)
25 y_hat = beta_1_hat * x + beta_0_hat
26
27 # 6. 結果の表示
28 print(f"価格の平均:{x_mean:.4f}")
29 print(f"評価の平均:{y_mean:.4f}")
30 print(f"傾き:{beta_1_hat:.4f}")
31 print(f"切片:{beta_0_hat:.4f}")
32 print(f"最終的な回帰式:y_hat={beta_0_hat:.4f}+{beta_1_hat:.4f}*x")
```

---