

R² (决定系数)

R² (决定系数)

核心定义：衡量回归模型拟合优度的指标，反映因变量的变异中能被自变量解释的比例，取值范围为 $(-\infty, 1]$ 。公式：

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

其中：

- y_i ：真实值
- \hat{y}_i ：模型预测值
- \bar{y} ：真实值的平均值

关键解读

- 理想情况：** $R^2 = 1 \rightarrow$ 模型完美拟合所有数据，预测值与真实值完全一致。
- 基线模型：** $R^2 = 0 \rightarrow$ 模型预测能力等价于“直接用真实值的平均值作为预测结果”。
- 糟糕模型：** $R^2 < 0 \rightarrow$ 模型预测效果比“平均值基线”更差（常见于非线性数据强行拟合线性模型）。

适用场景

- 仅用于**回归问题**（预测连续值，如浓度、温度、产量）。
- 适合评估模型对数据的**整体解释力**，但对异常值敏感（异常值会显著拉低 R^2 ）。

局限性

说明：

- 增加自变量个数时， R^2 会**被动增大**（即使新增自变量无实际意义），需用**调整后** R^2 ($Adjusted R^2$) 修正：

$$R_{adj}^2 = 1 - \frac{(1 - R^2)(n - 1)}{n - p - 1}$$

其中 p 是自变量个数， n 是样本量。

R² (决定系数) 计算

通用数据示例

先定义真实值和预测值：

```
import numpy as np

# 示例数据: 真实值 y_true, 预测值 y_pred
y_true = np.array([2.5, 3.8, 4.2, 5.0, 6.5]) # 真实值
y_pred = np.array([2.3, 3.9, 4.0, 5.2, 6.3]) # 模型预测值
```

方式 1: 手动实现 (公式还原)

```
def calculate_r2(y_true, y_pred):
    # 步骤1: 计算真实值的平均值
    y_mean = np.mean(y_true)
    # 步骤2: 计算残差平方和 (SSR) 和总平方和 (SST)
    ss_res = np.sum((y_true - y_pred) ** 2) # 模型预测偏差
    ss_tot = np.sum((y_true - y_mean) ** 2) # 真实值与平均值的偏差
    # 步骤3: 计算R²
    r2 = 1 - (ss_res / ss_tot) if ss_tot != 0 else 0 # 避免分母为0
    return r2

# 调用计算
r2_manual = calculate_r2(y_true, y_pred)
print(f"手动计算 R² = {r2_manual:.4f}") # 输出: 0.9768 (示例结果)
```

方式 2: 调用 sklearn 库 (推荐)

```
from sklearn.metrics import r2_score

# 直接调用库函数
r2_sklearn = r2_score(y_true, y_pred)
print(f"sklearn 计算 R² = {r2_sklearn:.4f}") # 结果与手动一致
```