**به نام خدا**

**توضیح کد :**

*# نرمال‌سازی داده‌ها (برای همگرایی بهتر)*

X = (X - np.mean(X)) / np.std(X)

این کد یک تبدیل آماری مهم انجام می‌دهد که به آن **نرمال‌سازی استاندارد** یا **Z-Score Normalization** می‌گویند. بیایید خط به خط تحلیل کنیم:

X = np.array([1, 2, 3, 4, 5]) *# داده اصلی*

X = (X - np.mean(X)) / np.std(X) *# نرمال‌سازی*

1. **محاسبه میانگین (**np.mean(X)**)**:



* میانگین، مرکز توزیع داده‌هاست.

### ****محاسبه انحراف معیار (****np.std(X)****)****:



**انحراف معیار (Standard Deviation)** یکی از مهم‌ترین شاخص‌های پراکندگی در آمار است که میزان پراکندگی یا تغییرپذیری داده‌ها را نسبت به میانگین نشان می‌دهد. به بیان ساده، انحراف معیار مشخص می‌کند که داده‌ها به طور متوسط چقدر از مقدار میانگین فاصله دارند.

### ****تفریق میانگین (****X - mean****)****:

### [1,2,3,4,5]−3=[−2,−1,0,1,2]

* داده‌ها را حول صفر متمرکز می‌کند (Mean Centering).

### ****تقسیم بر انحراف معیار (****/ std****)****:



* داده‌ها را مقیاس‌پذیر می‌کند.

### ****چرا این تبدیل مهم است؟****

۱. **معنای Z-Score:**  
هر عدد نشان می‌دهد آن نقطه داده چند انحراف معیار از میانگین فاصله دارد.  
مثلاً ۱.41 یعنی: «این مقدار ۱.۴۱ انحراف معیار بالاتر از میانگین است».

۲. **مزایا در یادگیری ماشین:**

* الگوریتم‌هایی مانند SVM، رگرسیون خطی و شبکه‌های عصبی با **داده‌های نرمال‌شده** بهتر کار می‌کنند.
* از dominance ویژگی‌های با مقیاس بزرگتر جلوگیری می‌کند.
* به الگوریتم‌های مبتنی بر فاصله (مانند KNN) کمک می‌کند.

۳. **تفسیرپذیری:**

* مقادیر منفی: پایین‌تر از میانگین
* مقادیر مثبت: بالاتر از میانگین
* صفر: دقیقاً برابر میانگین

**توضیح کد :**

cost = np.mean((X\_b.dot(theta) - y.reshape(-1, 1))\*\*2)

این کد **تابع هزینه (Cost Function)** را برای رگرسیون خطی محاسبه می‌کند که به آن **Mean Squared Error (MSE)** می‌گویند. بیایید به دقت تحلیل کنیم:

### ****اجزای کد به صورت مرحله‌به‌مرحله:****

۱. X\_b.dot(theta):

* محاسبه پیش‌بینی‌های مدل : ŷ = θ₀ + θ₁x
* X\_b ماتریس طراحی (Design Matrix) با ستون ۱ برای ضریب اریب
* مثال: اگر X\_b = [[1, x₁], [1, x₂]] و theta = [θ₀, θ₁] باشد:

ŷ = [[1, x₁] \* [θ₀] = [θ₀ + θ₁x₁]

[1, x₂] [θ₁]] [θ₀ + θ₁x₂]]

۲. - y.reshape(-1, 1):

این دستور NumPy برای تغییر شکل (reshape) آرایه به یک **ستون (بردار عمودی)** استفاده می‌شود.

y = np.array([1, 2, 3, 4]) *# شکل: (4,)*

y\_reshaped = y.reshape(-1, 1) *# شکل: (4, 1)*

[[1]

[2]

[3]

[4]]

**توجه:**

* تفاضل پیش‌بینی‌ها از مقادیر واقعی (خطاها)
* reshape(-1, 1) اطمینان می‌دهد y به شکل ستونی (n×1) باشد

۳. \*\*2:

* مربع خطاها (برای حذف مقادیر منفی و تاکید بیشتر روی خطاهای بزرگ)

۴. np.mean():

* میانگین مربعات خطاها (MSE)

**توضیح کد :**

*# نرمال‌سازی داده‌ها*

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

StandardScaler یکی از مهم‌ترین ابزارهای **پیش‌پردازش داده** در یادگیری ماشین است که داده‌ها را استانداردسازی می‌کند. این فرآیند برای بسیاری از الگوریتم‌ها (مثل SVM، رگرسیون خطی، KNN) ضروری است.

### ۱. ****نحوه عملکرد:****

هر ویژگی را به صورت مستقل طوری تبدیل می‌کند که:

* میانگین (mean) = ۰
* انحراف معیار (std) = ۱

فرمول محاسبه:



#### **پس از استانداردسازی:**

[[-1.2247 -1.2247]

[ 0. 0. ]

[ 1.2247 1.2247]]

### ****تفسیر نتایج:****

* هر ستون میانگین ≈ ۰ و انحراف معیار ≈ ۱ دارد
* مقادیر منفی: پایین‌تر از میانگین
* مقادیر مثبت: بالاتر از میانگین

### ****مقایسه با دیگر Scalerها****

| **Scaler** | **فرمول** | **مناسب برای** |
| --- | --- | --- |
| StandardScaler | (X−μ)/σ(*X*−*μ*)/*σ* | داده‌های با توزیع نرمال |
| MinMaxScaler | (X−Xmin)/(Xmax−Xmin)(*X*−*Xmin*​)/(*Xmax*​−*Xmin*​) | داده‌های با محدوده مشخص |
| RobustScaler | (X−median)/IQR(*X*−median)/*IQR* | داده‌های با outlier |