**به نام خدا**

نکات کلیدی تغییرات:

1. **استفاده از**SGDRegressor:
   * پیاده‌سازی بهینه‌شده Gradient Descent در scikit-learn
   * پارامتر learning\_rate='constant' و eta0=0.1 معادل نرخ یادگیری 0.1 در کد شما
2. **نرمال‌سازی داده‌ها**:
   * استفاده از StandardScaler برای نرمال‌سازی Z-Score به جای محاسبه دستی
3. **محاسبه هزینه**:
   * محاسبه MSE در هر تکرار با partial\_fit
   * ذخیره تاریخچه هزینه برای رسم نمودار
4. **خروجی مشابه**:
   * نمایش ضرایب نهایی (θ0 و θ1)
   * رسم نمودار کاهش هزینه بر اساس تکرارها

این کد همان منطق Gradient Descent شما را با استفاده از ابزارهای حرفه‌ای scikit-learn پیاده‌سازی می‌کند.

**توضیحات کد :**

model.partial\_fit(X\_scaled, y)

در کتابخانه scikit-learn، متد partial\_fit() یک قابلیت کلیدی برای آموزش **تدریجی (Incremental Learning)** مدل‌ها است. بیایید به زبان ساده بررسی کنیم:

### 🎯 ****مفهوم****partial\_fit()****:****

* این متد به مدل اجازه می‌دهد که **به صورت قطعه‌قطعه (بچ‌به‌بچ) آموزش ببیند**، بدون نیاز به بارگذاری تمام داده‌ها در حافظه.
* برخلاف fit() که همه داده‌ها را یکجا پردازش می‌کند، partial\_fit() برای **داده‌های حجیم** یا **جریان‌های داده (Streaming Data)** ایده‌آل است.

### ****تفاوت****partial\_fit()****با****fit():

| **ویژگی** | partial\_fit() | fit() |
| --- | --- | --- |
| **حجم داده** | مناسب برای داده‌های بزرگ/جریانی | مناسب برای داده‌های کوچک |
| **حافظه مورد نیاز** | کم (داده‌ها را بچ‌به‌بچ پردازش می‌کند) | زیاد (همه داده‌ها یکجا بارگذاری می‌شوند) |
| **تعداد فراخوانی** | چندباره (تدریجی) | یکباره |
| **کاربردهای رایج** | یادگیری آنلاین، داده‌های حجیم | داده‌های کوچک و ایستا |

### ****چرا در کد Gradient Descent از****partial\_fit()****استفاده شد؟****

for i in range(100):

model.partial\_fit(X\_scaled, y) *# آموزش تدریجی در هر تکرار*

* **شبیه‌سازی رفتار Gradient Descent دستی**: در هر تکرار (epoch)، مدل با همان داده‌ها دوباره آموزش می‌بیند (مثل کد اصلی شما).
* **ذخیره تاریخچه هزینه**: با فراخوانی partial\_fit() در هر تکرار، می‌توانیم MSE را محاسبه و ذخیره کنیم.

### 📌 ****نکات مهم درباره****partial\_fit():

1. **اولین فراخوانی**:
   * باید حتماً اولین بار با تمام ویژگی‌ها فراخوانی شود تا مدل وزن‌ها را مقداردهی کند.
2. **پیش‌نیازها**:
   * فقط برخی مدل‌های scikit-learn این متد را پشتیبانی می‌کنند (مثل SGDRegressor, MultinomialNB).
   * برای مدل‌هایی مثل RandomForest یا SVM موجود نیست.

### ****جمع‌بندی کاربرد در کد شما****:

* **هدف**: شبیه‌سازی دقیقاً همان رفتار Gradient Descent دستی که در کد اصلی بود.
* **مزیت**: استفاده از پیاده‌سازی بهینه‌شده scikit-learn به جای نوشتن حلقه‌های دستی.
* **توجه**: در عمل برای داده‌های کوچک (مثل مثال شما) معمولاً از fit() استفاده می‌شود.