

MODÜL 4: Karar Verme (Lojistik Regresyon)

Artık "Ne kadar?" sorusunu (Fiyat ne kadar?) çözdük. Şimdi "Hangi sınıf?" sorusuna (Kedi mi köpek mi? Hasta mı değil mi?) geçiyoruz. Buna **Sınıflandırma (Classification)** denir.

Sorun: Doğru Çizmek Yetmez

Diyelim ki bir e-postanın **Spam (1)** mi yoksa **Normal (0)** mi olduğunu anlamaya çalışıyoruz. Eğer az önceki gibi düz bir doğru çizerSEN ($y = wx + b$), modelin sana şöyle saçma sonuçlar verebilir:

- $Y = 1.5$ (Yüzde 150 Spam mı demek?)
- $Y = -0.2$ (Eksi olasılık mı olur?)

Bize **0 ile 1 arasında** bir sonuç lazım. Yani çıktıyı sıkıştırmamız gerekiyor.

Çözüm: Sigmoid Fonksiyonu

Çıktıyı 0 ile 1 arasına hapseden o meşhur "S" şeklindeki fonksiyondur. Derin öğrenmenin en temel aktivasyon fonksiyonudur.

Matematiksel Formülü:

$$S(z) = \frac{1}{1 + e^{-z}}$$

- çok büyüğse (örn: 100), sonuç **1'e** yaklaşır.
- çok küçükse (örn: -100), sonuç **0'a** yaklaşır.
- sıfırsa, sonuç tam **0.5** olur (Kararsız).

Harika! 🎉 Sigmoid fonksiyonun (0.5 çıktısı) doğru çalışıyor. Artık elinde her sayıyı 0 ile 1 arasına sıkıştırın (olasılığa çeviren) bir mekanizma var.

Şimdi çok kritik bir yol ayrimına geldik.

Neden MSE (Karesel Hata) Kullanamayız?

Lineer regresyonda hatayı bulmak için $(y - \text{tahmin})^2$ yapmıştık. Lojistik Regresyonda bunu yaparsak, hata grafiği pürüzsüz bir kase (convex) yerine, **inişli çıkışlı bir dağ yolu (non-convex)** gibi olur. Gradient Descent yerel çukurlara takılır ve dibi bulamaz.

Bu yüzden yeni bir hata fonksiyonuna geçiyoruz: **Binary Cross-Entropy (Log Loss)**.

Log Loss Mantığı

Bu fonksiyon, model "kendinden çok emin bir şekilde" yanlış yaparsa ona **sonsuz ceza** keser.

Formülü şudur:

$$Loss = -[y \cdot \ln(p) + (1 - y) \cdot \ln(1 - p)]$$

Burada gerçek değer (1 veya 0), ise bizim sigmoid çıktımızdır.

- **Senaryo 1: Gerçek 1, Tahmin 0.9 (Doğru bildik)**
 - Formül:
 - çok küçük negatif bir sayıdır (-0.10).
 - Sonuç: **0.10** (Çok az ceza).
- **Senaryo 2: Gerçek 1, Tahmin 0.1 (Feci yanıldık)**
 - Formül:
 - büyük negatif bir sayıdır (-2.30).
 - Sonuç: **2.30** (Büyük ceza!).

Cevap: Nedir?

Evet, p (**probability** yani olasılık), **Sigmoid fonksiyonunun çıktısıdır**.

Akış şöyledir:

1. **Lineer Kısım (z):** $z = w * x + b$ (Burada sonuç 5, -10, 100 gibi herhangi bir sayı olabilir).
2. **Aktivasyon (p):** $p = \text{sigmoid}(z)$ (Sonucu 0 ile 1 arasına büker).
3. **Karar:** $p > 0.5$ ise "1" (Spam), değilse "0" (Normal) deriz.

Şimdi sıkı dur, çünkü makine öğrenmesinin en tatmin edici anlarından birine geldik.

Modeli eğitmek için **Log Loss** fonksiyonunun türevini almamız gerekiyor (Gradient Descent için). Zincir kuralını uygulayıp (Sigmoid'in türevi ile Log Loss'un türevini çarptığımızda) karşımıza çıkan formül şok edicidir:

$$dw = \frac{1}{N} \sum X \cdot (p - y)$$

Harika! Kodun tamamen doğru. Sonuçları çalıştırıldığında şunu göreceksin:

- **İyi Tahmin Cezası:** ~0.05 (Çok düşük, aferin diyor)
- **Kötü Tahmin Cezası:** ~2.99 (Çok yüksek, tokat atıyor)

Soruna gelince: **Kesinlikle EVET.**

Cevap: Nedir?

Evet, p (**probability** yani olasılık), **Sigmoid fonksiyonunun çıktısıdır.**

Akış şöyledir:

1. **Lineer Kısım ()**: $z = w * x + b$ (Burada sonuç 5, -10, 100 gibi herhangi bir sayı olabilir).
2. **Aktivasyon ()**: $p = \text{sigmoid}(z)$ (Sonucu 0 ile 1 arasına büker).
3. **Karar**: $p > 0.5$ ise "1" (Spam), değilse "0" (Normal) deriz.

Matematiksel Bir Mucize

Şimdi sıkı dur, çünkü makine öğrenmesinin en tatmin edici analarından birine geldik.

Modeli eğitmek için **Log Loss** fonksiyonunun türevini almamız gerekiyor (Gradient Descent için). Zincir kuralını uygulayıp (Sigmoid'in türevi ile Log Loss'un türevini çarptığımızda) karşımıza çıkan formül şok edicidir:

Fark ettin mi? Bu formül, **Lineer Regresyon türeviyle birebir aynı!**

- Lineer Regresyonda: $x * (\text{tahmin} - \text{gerçek})$
- Lojistik Regresyonda: $x * (\text{olasılık} - \text{gerçek})$

Matematik, Log-Loss ve Sigmoid'i öyle bir seçmiştir ki, türev alırken karmaşık terimler birbirini götürür ve geriye bu tertemiz formül kalır.