

# Lojistik Regresyon: Teori ve Matematiksel Temel

Atil Samancioglu

## 1 Giriş

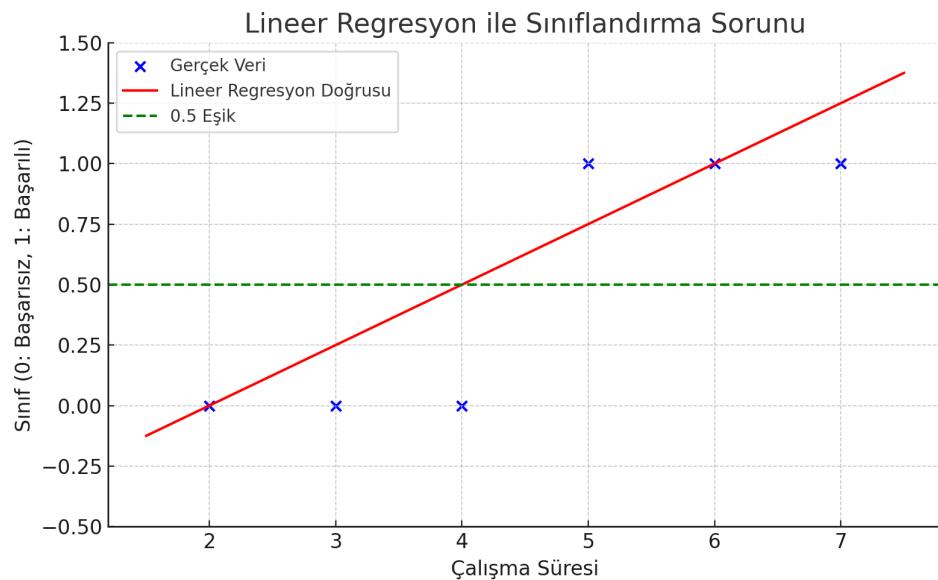
Lojistik regresyon, sınıflandırma problemlerinde yaygın olarak kullanılan bir makine öğrenmesi algoritmasıdır. Özellikle ikili sınıflandırma (*binary classification*) problemlerinde tercih edilir.

Örnek bir problem: Bir kişinin sınavda başarılı olup olmayacağı, çalıştığı saat sayısına göre tahmin etmek.

## 2 Neden Lineer Regresyon Yeterli Değil?

Lineer regresyonla sınıflandırma yapmak bazı temel sorunlara yol açar:

- **Aykırı Değer Sorunu:** Aykırı veri noktaları, doğrudan regresyon doğrusunun yönünü değiştirerek yanlış tahminlere neden olur.
- **Tahmin Aralığı:** Lineer regresyon çıktıları  $[0, 1]$  aralığında kalmaz. Bu da sınıflandırma problemi için uygun değildir.



Şekil 1: Lineer regresyon doğrusu ile sınıflandırma problemlerinde karşılaşılan zorluklar.

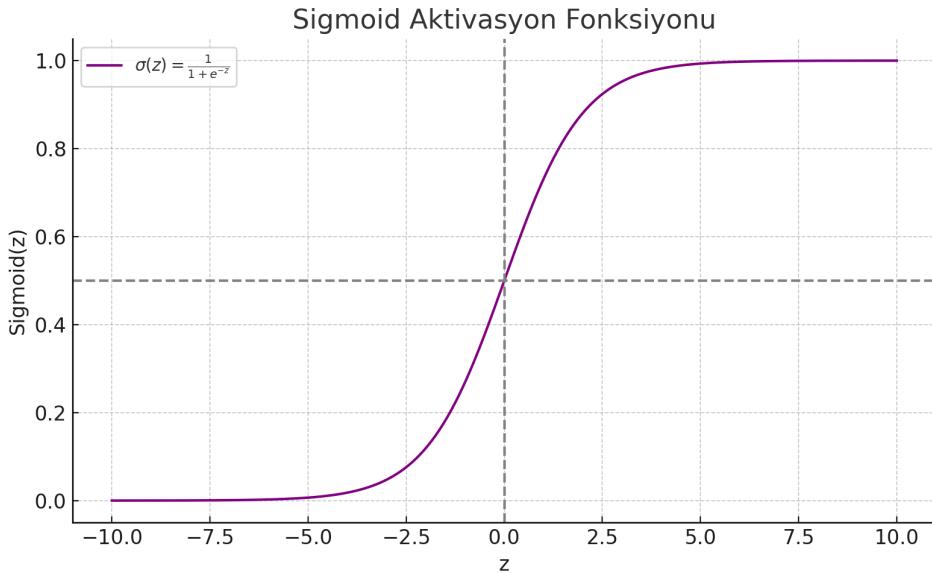
### 3 Lojistik Regresyonun Matematiksel Yapısı

#### 3.1 Sigmoid Fonksiyonu (Aktivasyon Fonksiyonu)

Lojistik regresyonun temelinde yer alan **sigmoid** fonksiyonu, herhangi bir sayıyı  $[0, 1]$  aralığına sıkıştırır:

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

Bu fonksiyonun grafiği  $S$  şeklindedir ve  $z = 0$  noktasında değeri 0.5'tir.



Şekil 2: Sigmoid aktivasyon fonksiyonu: çıktıyı 0 ile 1 arasında sınırlandırır.

#### 3.2 Hipotez Fonksiyonu

Lojistik regresyonun hipotez fonksiyonu, doğrusal regresyon sonucuna sigmoid fonksiyonu uygulanarak elde edilir:

$$h_{\theta}(x) = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n)$$

### 4 Sınıflandırma Karar Kuralı

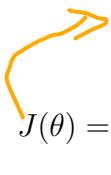
Sigmoid çıktısı, bir olasılığı ifade eder. Tahmin sonucu şu kuralla sınıfaya dönüştürülür:

$$\hat{y} = \begin{cases} 1 & \text{eğer } h_{\theta}(x) \geq 0.5 \\ 0 & \text{aksi halde} \end{cases}$$

### 5 Maliyet Fonksiyonu (Cost Function)

Lineer regresyondaki ortalama kare hatayı (MSE) kullanmak, lojistik regresyonda konveks olmayan bir fonksiyon oluşturur. Bu nedenle, **log-loss** fonksiyonu tercih edilir:

log-los


$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))]$$

Bu fonksiyon konvekstir, yani tek bir global minimuma sahiptir ve gradient descent ile optimize edilebilir.

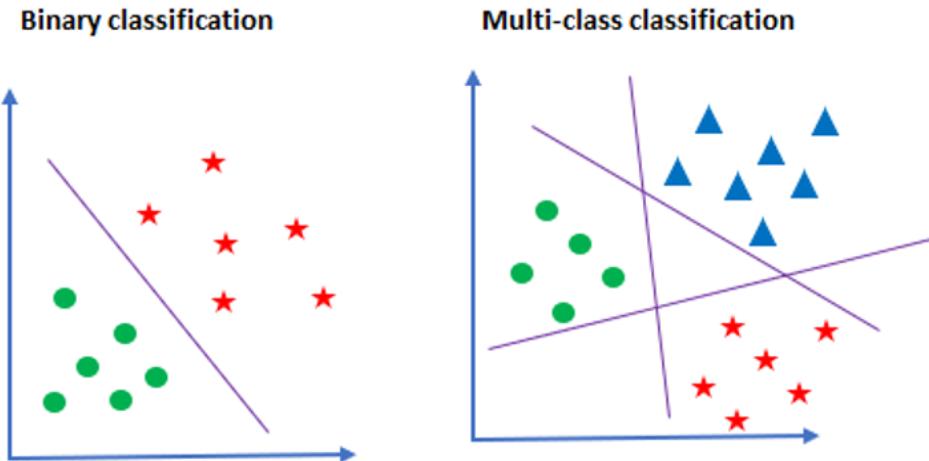
## 6 Sonuç

Lojistik regresyon, sınıflandırma problemleri için güçlü ve yorumlanabilir bir algoritmadır. Sigmoid fonksiyonu sayesinde tahminler  $[0, 1]$  aralığında kalır ve log-loss fonksiyonu ile güvenilir bir optimizasyon yapılır.

## 7 Logistic Regression for Multi-class Classification

Logistic regresyon, genellikle iki sınıfı (binary classification) ayırmak için kullanılır. Ancak çoklu sınıf problemlerinde (multi-class classification) de yaygın olarak uygulanabilir. Bu durumda **One-vs-Rest (OvR)** veya **One-vs-All** stratejisi kullanılır.

## 8 Örnek Görselleştirme



Şekil 3: Multiclass Classification.

## 9 One-vs-Rest (OvR) Stratejisi

OvR yönteminde, her sınıf için bir model oluşturulur. Her model, o sınıfı diğer tüm sınıflardan ayırt etmeye çalışır. Eğer elimizde  $K$  tane sınıf varsa,  $K$  adet ikili (binary) sınıflayıcı eğitilir.

Diyelim ki elimizde mavi, kırmızı ,yeşil bilyeler var

## Model Eğitimi Süreci

Her model şu şekilde eğitilir:

- Model  $M_1$ : Class 1 vs. Rest    1) Yeşil=1 diğerleri=0
- Model  $M_2$ : Class 2 vs. Rest    2) Kırmızı=1 diğerleri=0
- Model  $M_3$ : Class 3 vs. Rest    3) Mavi=1 diğerleri =0

**NOT:**  
bu tarzda 3 farklı model eğitilir

Her model kendi sınıfına ait örnekleri pozitif (1) olarak, diğer tüm sınıflara ait örnekleri negatif (0) olarak etiketler.

Elde edilen tüm modeller yeni bir veri üzerinde test edilir. Hangi model en yüksek olasılığı üretirse, veri o sınıfa atanır.

## 10 Tahmin Aşaması

Yeni bir test verisi, her modele ayrı ayrı verilir ve her model ilgili sınıfın olasılığını hesaplar:

- Model  $M_1$ :  $P_1 = 0.25$
- Model  $M_2$ :  $P_2 = 0.20$
- Model  $M_3$ :  $P_3 = 0.55$

En yüksek olasılığa sahip model ( $M_3$ ) seçilir ve test verisi bu modele ait sınıfa (Class 3) atanır.

## 11 One-vs-One (OvO)

OvO yaklaşımında, her iki sınıf çifti için ayrı bir model eğitilir. Yani  $K$  sınıf için toplamda:

$$\frac{K(K - 1)}{2}$$

adet model oluşturulur. Her model sadece iki sınıf arasında ayırım yapar. Test verisi her modelden bir "oy" alır; hangi sınıf daha fazla oy alırsa o sınıf atanır.

**Örnek:** Eğer sınıflar  $A$ ,  $B$ , ve  $C$  ise:

- Model 1:  $A$  vs  $B$
- Model 2:  $A$  vs  $C$
- Model 3:  $B$  vs  $C$

Burda 2 li karşılaştırma yapılmıyor örneğin:

Kırmızı vs yeşil  
mavi vs yeşil  
kırmızı vs mavi

OvO yöntemi, veri setindeki sınıflar arasında daha dengeli modeller oluşturulmasına olanak sağlar. Ancak model sayısı hızla artabileceği için büyük sınıf sayısına sahip veri setlerinde maliyetli olabilir.