# Regresyon

Atil Samancioglu

## 1 Basit Doğrusal Regresyona Giriş

Basit Doğrusal Regresyon, gözetimli makine öğrenmesinde temel algoritmalardan biridir. Bu yöntem, biri bağımsız (girdi) ve diğeri bağımlı (çıktı) olmak üzere iki sürekli değişken arasındaki ilişkiyi modellendirir. Sadeliğine rağmen, burada öğreneceğiniz kavramlar daha karmaşık modellerde, örneğin yapay sinir ağlarında da kullanılır.

## 2 Problemi Anlamak: Neden Tahmin Yapmak İsteriz?

Hayatın birçok alanında geleceğe dair tahminler yapmak isteriz:

- Bir reklam kampanyası yaparsak satışlar ne kadar artar?
- Bir evin büyüklüğüne göre fiyatı ne olur?
- Hava sıcaklığı arttıkça dondurma satışı nasıl değişir?

Bu tür soruları yanıtlamak için elimizdeki verilere bakıp **geleceği öngören modeller** kurmak isteriz.

Bu bölümde bu hedefe ulaşmak için kullanılan en temel yöntemlerden biri olan **Basit Doğrusal Regresyon** ile tanışacağız.

## 3 Gerçek Hayattan Örnek: Reklam Harcaması ve Satış Geliri

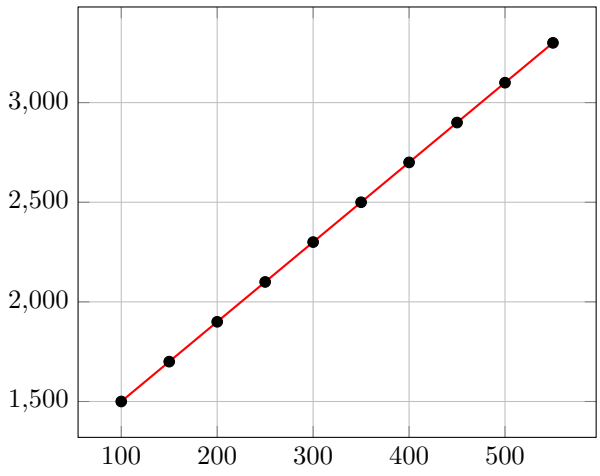
Bir işletmenin aşağıdaki verileri kaydettiğini düşünelim:

| Reklam Harcaması ($) | Satış Geliri ($) |
|---|---|
| 100 | 1500 |
| 150 | 1700 |
| 200 | 1900 |
| 250 | 2100 |
| 300 | 2300 |
| 350 | 2500 |
| 400 | 2700 |
| 450 | 2900 |
| 500 | 3100 |
| 550 | 3300 |

Peki, 475$'lık bir reklam harcaması yaptığımızda ne kadar gelir elde edebiliriz? İşte bunu tahmin etmek istiyoruz.

## 4 Veriyi Görselleştirmek

Önce veriye bakalım:




Şekil 1: Veri noktaları ve en iyi uyum doğrusu.

**Gözlem:** Reklam harcaması arttıkça satış gelirinin de arttığını görüyoruz. Bu ilişkiyi bir doğru (düz çizgi) ile modelleyebiliriz.

## 5 Model Kurmak: Doğrusal Bir Denklem

Bir doğruyu ifade etmek için:

$$\hat{y} = \theta_0 + \theta_1 x$$

kullanırız. Burada:

- $\theta_0$: Doğrunun y eksenini kestiği nokta (başlangıç değeri)
- $\theta_1$: Eğim (harcama arttıkça satışın ne kadar arttığı)
- $x$: Girdi (reklam harcaması)
- $\hat{y}$: Tahmin edilen çıktı (satış geliri)

Örnek parametreler:

$$\theta_0 = 1100, \quad \theta_1 = 4$$

Bu durumda:

$$\hat{y} = 1100 + 4x$$

## 6 Modeli Kullanarak Tahmin Yapmak

450$ reklam harcaması yaparsak:

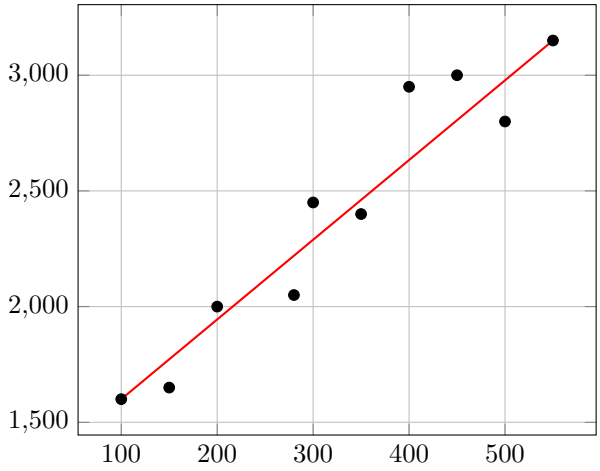
$$\hat{y} = 1100 + 4 \times 450 = 2900$$

Yani 450$'lık reklam harcaması sonucunda 2900$ satış bekliyoruz.

## 7 Gerçek Dünya: Her Şey Mükemmel Mi?

Ne yazık ki gerçek hayatta veriler mükemmel bir doğru üzerinde durmaz. Bunu aşağıdaki örnekle görebiliriz:

| Reklam Harcaması ($) | Satış Geliri ($) |
|---|---|
| 100 | 1600 |
| 150 | 1650 |
| 200 | 2000 |
| 280 | 2050 |
| 300 | 2450 |
| 350 | 2400 |
| 400 | 2950 |
| 450 | 3000 |
| 500 | 2800 |
| 550 | 3150 |




Şekil 2: Gürültü içeren veriler ve en iyi uyum doğrusu.

**Gözlem:** Veriler doğruya yakın ama birebir üzerinde değil. Bu yüzden **hata** kavramı doğar.

## 8 Hataları Ölçmek: Maliyet Fonksiyonu

Bir tahminin ne kadar doğru olduğunu ölçmek için:

$$\text{Hata} = y - \hat{y}$$

ve tüm hataların karesinin ortalamasını alırız:

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

Bu değere **Mean Squared Error (MSE)** denir.
Modelin amacı: MSE'yi mümkün olan en küçük değere indirmek.

## 9 Modeli İyileştirmek: Gradient Descent

Peki en düşük hataya ulaşmak için parametreleri ($\theta_0$, $\theta_1$) nasıl değiştireceğiz?

# 10 Modeli İyileştirmek: Gradient Descent

Modelimizin tahminleri mükemmel olmayabilir. Tahminle gerçek değer arasındaki farkı ölçerek bir **hata** (maliyet) hesaplıyoruz.

Bu hatayı mümkün olduğunca küçük yapmak istiyoruz. Peki nasıl?

İşte burada **Gradient Descent** (Gradyan İnişi) adlı algoritma devreye giriyor.

## Neyi Değiştiriyoruz?

Bizim elimizde ayarlanabilir iki parametre var:

- $\theta_0$: Doğrunun y-kesişim noktası (başlangıç değeri)
- $\theta_1$: Doğrunun eğimi (satışların reklama nasıl tepki verdiği)

Modelimizin doğrusu bu parametrelere bağlı:

$$\hat{y} = \theta_0 + \theta_1 x$$

Eğer $\theta_0$ ve $\theta_1$ değerlerini doğru seçersek, tahminlerimiz daha iyi olur ve hata azalır.

## Örnek:

Varsayalım başlangıçta:

$$\theta_0 = 500, \quad \theta_1 = 2$$

olsun.

Bu değerlerle modelimiz yanlış tahminler yapıyor ve hata yüksek çıkıyor.

Biz $\theta_0$ ve $\theta_1$ değerlerini küçük küçük değiştirerek (güncelleyerek) hatayı azaltmaya çalışacağız.

## Bunu Nasıl Yapacağız?

Bu süreç şöyle işler:

1. Mevcut $\theta$ değerleri ile modelimizin hata (maliyet) değerini hesaplarız.
2. Hatayı azaltmak için hangi yöne gitmemiz gerektiğini buluruz (eğim).
3. Küçük bir adım atarız ve $\theta$ değerlerini güncelleriz.
4. Bu adımları tekrarlarız. Her seferinde hata biraz daha azalır.

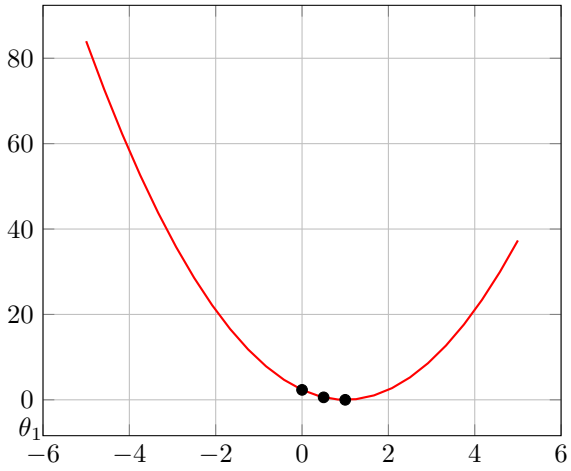
Bu adımlar, "yokuştan aşağı inmek" gibi düşünülebilir.

**Analojik düşünün:** Karanlık bir dağın tepesindeyiz. Gözümüz kapalı, sadece bulunduğumuz noktanın eğimine (ne kadar dik ve hangi yöne meyilli olduğuna) bakarak en aşağıya inmek istiyoruz.

Her adımda: - Eğer eğim pozitifse (yokuş yukarıysa) geriye doğru adım atıyoruz. - Eğer eğim negatifse (yokuş aşağıysa) ileriye doğru adım atıyoruz.

Ve küçük küçük inerek en dip noktaya (en düşük hata noktasına) ulaşıyoruz.

**Görsel:**




Şekil 3: Maliyet fonksiyonu ve minimum hata noktası.

Grafikte, yatay eksende $\theta_1$ değerleri ve dikey eksende modelimizin hatası (MSE) var. Hedefimiz:

$$\theta_1 = 1$$

değerine ulaşmak çünkü hata burada minimum (sıfır) oluyor.

### Gradient Descent Güncelleme Formülü

Bu inişi yapmak için kullandığımız güncelleme formülü:

$$\theta := \theta - \alpha \cdot \frac{\partial J(\theta)}{\partial \theta}$$

Burada:

- $\alpha$: Öğrenme oranı (ne kadar büyük adım atacağımız)
- $\frac{\partial J(\theta)}{\partial \theta}$: O noktadaki eğim (ne tarafa inmeli?)

**Önemli:**

- Çok büyük adımlar atarsak (büyük $\alpha$), hedefi aşabiliriz.
- Çok küçük adımlar atarsak (küçük $\alpha$), çok yavaş ineriz.

Bu yüzden doğru bir öğrenme oranı seçmek çok önemlidir.

—

## 11 Öğrendiklerimizi Özetleyelim

- Gerçek verilerden hareketle geleceği tahmin etmek isteriz.
- Basit doğrusal regresyon bu tahminler için temel bir yöntemdir.
- Veriler her zaman kusursuz değildir → hata kavramı doğar.
- Hataları minimize etmek için Gradient Descent kullanılır.

Bu temel prensipler, tüm makine öğrenmesi algoritmalarının yapı taşlarını oluşturur.

# 12 Çoklu Doğrusal Regresyon (Multiple Linear Regression)

Şimdiye kadar yalnızca **bir girdi (bağımsız değişken)** içeren regresyon modelleriyle çalıştık. Buna **Basit Doğrusal Regresyon** denir. Peki ya birden fazla girdimiz varsa?

İşte bu durumda kullandığımız modele **Çoklu Doğrusal Regresyon** (*Multiple Linear Regression*) denir.

## Basit Tekrarlama: Tek Girdili Model

Hatırlayalım:

$$\hat{y} = \theta_0 + \theta_1 x$$

Burada:

- $x$: Bağımsız değişken (örneğin: kilo)
- $\hat{y}$: Tahmin edilen bağımlı değişken (örneğin: boy)
- $\theta_0$: Y ekseni kesişimi (intercept)
- $\theta_1$: Eğim (slope)

—

## Yeni Durum: Birden Fazla Özellik (Feature)

Diyelim ki bir evin fiyatını tahmin etmek istiyoruz. Elimizde şu bilgiler var:

- $x_1$: Oda sayısı
- $x_2$: Ev büyüklüğü (metrekare)
- $x_3$: Lokasyon (sayısal kodlanmış)

Modelimizin tahmin formülü şu hale gelir:

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$

**Genel hali**:

$$\hat{y} = \theta_0 + \sum_{j=1}^{n} \theta_j x_j$$

Burada $n$, toplam özellik (feature) sayısını gösterir.

—

## Parametreler Ne Anlama Geliyor?

Her bir $\theta_j$:
- İlgili özelliğin ev fiyatı üzerindeki etkisini ifade eder.
- Modelin bu katsayıları öğrenmesi gerekir.
- $\theta_0$ her zaman sabit terimdir (intercept), tüm özelliklerden bağımsızdır.

—

## Gradient Descent Bu Durumda Nasıl Çalışır?

Basit regresyonda sadece $\theta_0$ ve $\theta_1$ vardı. Ama çoklu regresyonda $\theta_2, \theta_3, \ldots$ gibi birçok katsayı vardır.
Her bir parametre için ayrı ayrı güncelleme yapılır:

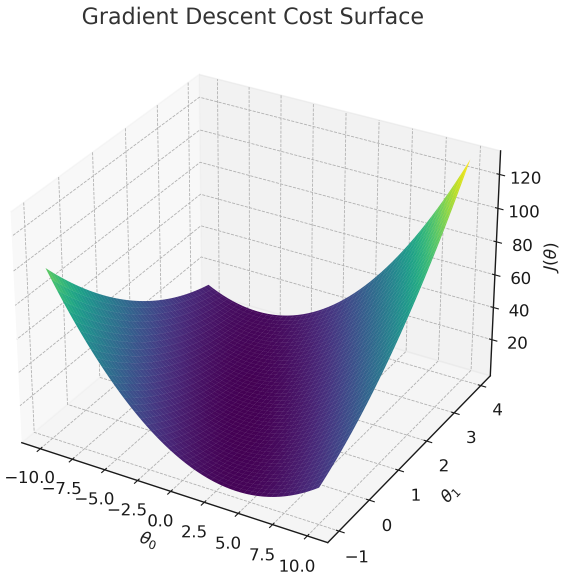
$$\theta_j := \theta_j - \alpha \cdot \frac{1}{m} \sum_{i=1}^{m} \left( \hat{y}^{(i)} - y^{(i)} \right) x_j^{(i)} \quad \text{(her } j \text{ için)}$$

- $j = 0$ olduğunda $x_0 = 1$ kabul edilir (sabit terim için)
- Bu şekilde her $\theta_j$ gradyan inişiyle optimize edilir

—

## 3D Perspektif: Daha Karmaşık Yüzey

- Tek değişkenli regresyonda maliyet fonksiyonu bir eğriydi (2D grafik)
- İki veya daha fazla parametre varsa, bu artık bir yüzeydir (3D grafik)
- Yüzeyin çukur noktası, **global minimum** noktasıdır




Şekil 4: Gradient Descent ile 3D maliyet yüzeyi.

—

## Özet: Simple vs Multiple Linear Regression

- Sadece 1 özellik → **Basit Doğrusal Regresyon**
- 2 veya daha fazla özellik → **Çoklu Doğrusal Regresyon**

Model ve öğrenme mantığı aynıdır. Sadece:

- Katsayı sayısı artar
- Hesaplama karmaşıklığı artar
- Ama prensip aynı kalır: **Maliyet fonksiyonunu minimize et!**