

---

# Regresyon

**Veri Bilimine Giriş**

**Dr. Öğr. Üyesi Fatma Zehra SOLAK**



# Doğrusal Regresyon ve Kuzenleri

---

- Basit Doğrusal Regresyon
- Çoklu Doğrusal Regresyon
- Temel Bileşen Regresyonu
- Kısmi En Küçük Kareler Regresyonu
- Ridge Regresyon
- Lasso Regresyon
- Elastic Net Regresyonu
- Her Model İçin:
  - Model
  - Tahmin
  - Model Optimizasyonu

---

# Basit Doğrusal Regresyon



# Basit Doğrusal Regresyon

---

Temel amaç, bağımlı ve bağımsız değişken arasındaki ilişkiyi ifade eden doğrusal fonksiyonu bulmaktır.

	y1	x1
1	8.04	10
2	6.95	8
3	7.58	13
4	8.81	9
5	8.33	11
6	9.96	14
7	7.24	6
8	4.26	4
9	10.84	12
10	4.82	7
11	5.68	5



# Basit Doğrusal Regresyon

Anakitle teorik gösterim:  $Y = \beta_0 + \beta_1 X + \varepsilon$

Örneklem gerçek değerler:  $y_i = b_0 + b_1 x_i + e_i$

Tahmin modeli:  $\hat{y}_i = b_0 + b_1 x_i$

$\beta_0$  = Doğrunun y eksenini kestiği nokta

$\beta_1$  = Doğrunun eğimi

$\varepsilon$  = Hata terimi

X: bağımsız değişkenler

Y: bağımlı değişkenler

(Sadece bir tane bağımsız değişken söz konusudur)

## En Küçük Kareler


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

(Hata kareler toplamını minimize edecek şekilde katsayıları tahmin etmeye çalışacağız)


$b_0$  ve  $b_1$  katsayılarını bulmaya çalışarak yeni gelecek örneklemelerin bağımsız değişken değerini bildiğimizde bağımlı değişken değeri nedir tahmin edeceğiz.



# Basit Doğrusal Regresyon

$$SSE = \sum_{i=1}^n (y_i - (b_0 + b_1 x_i))^2$$


$$b_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

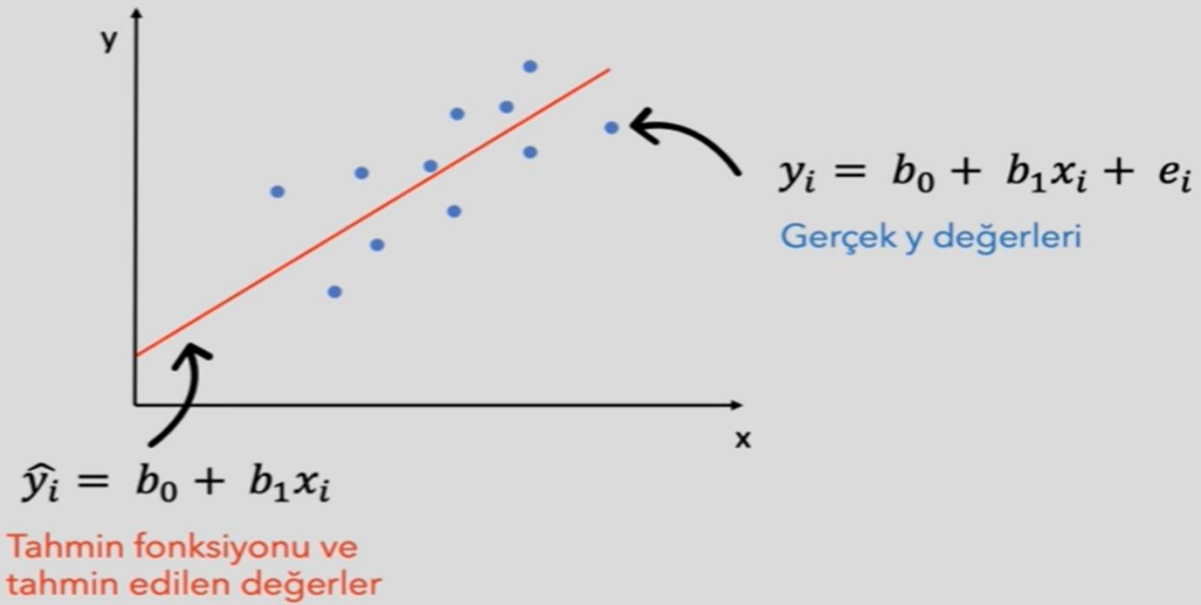
$$b_0 = \bar{y} - b_1 \bar{x}$$




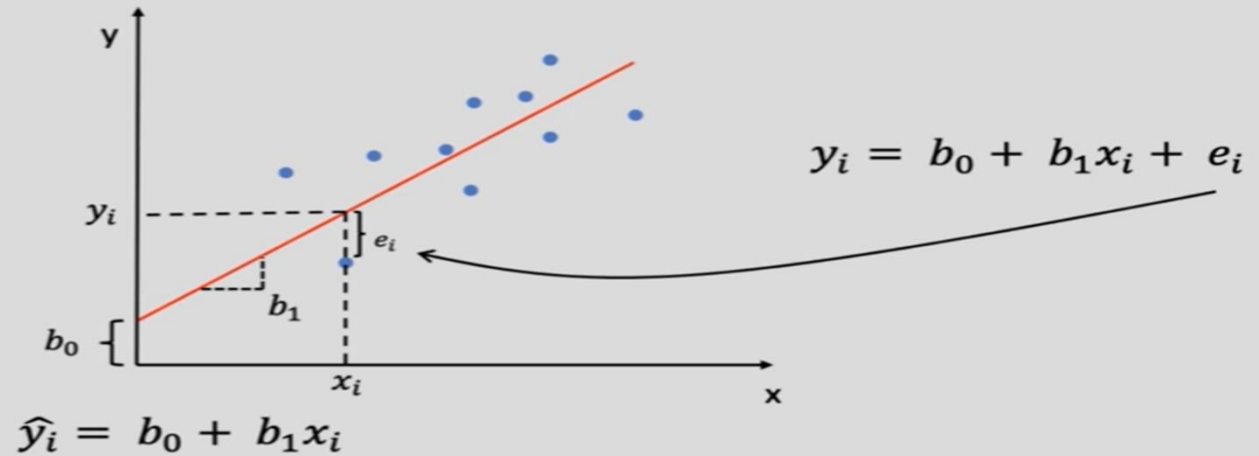
Bağımlı değişkenin ortalaması



## Basit Doğrusal Regresyon Geometrik Gösterim



## Basit Doğrusal Regresyon Geometrik Gösterim



# Çoklu Doğrusal Regresyon

---

**Temel amaç, bağımlı ve bağımsız değişkenler arasındaki ilişkiyi ifade eden doğrusal fonksiyonu bulmaktır.**





# Çoklu Doğrusal Regresyon

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_j X_{ij} + \cdots + \beta_p X_{ip} + \varepsilon_i$$

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

$$\hat{\beta} = (X^T \cdot X)^{-1} X^T \cdot Y$$



# Çoklu Doğrusal Regresyon

## Doğrusal Regresyonun Varsayımları

- Hatalar normal dağılır.
- Hatalar birbirinden bağımsızdır ve aralarında otokorelasyon yoktur.
- Her bir gözlem için hata terimleri varyansları sabittir.
- Değişkenler ile hata terimi arasında ilişki yoktur.
- Bağımsız değişkenler arasında çoklu doğrusal ilişki problemi yoktur.



# Çoklu Doğrusal Regresyon

## Regresyon Modellerinin Avantaj ve Dezavantajları

- ✓ İyi anlaşılırsa diğer tüm ML ve DL konuları çok rahat kavranır.
- ✓ Doğrusallık nedensellik yorumları yapılabilmesini sağlar, bu durum aksiyoner ve stratejik modelleme imkanı verir.
- ✓ Değişkenlerin etki düzeyleri ve anlamlılıkları değerlendirilebilir.
- ✓ Bağımlı değişkendeki değişkenliğin açıklanma başarısı ölçülebilir.
- ✓ Model anlamlılığı değerlendirilebilir.
- ❖ Varsayımları vardır.
- ❖ Aykırı gözlemlere duyarlıdır.



# Doğrusal Olmayan Regresyon Modelleri

---

- K-En Yakın Komşu (KNN)
- Destek Vektör Regresyonu (SVR)
- Çok Katmanlı Algılayıcılar (ANN)
- Classification and Regression Trees (CART)
- Bagging (Bootstrap Aggregation)
- Random Forests (RF)
- Gradient Boosting Machines (GBM)
- Extreme Gradient Boosting (XGBoost)
- LightGBM
- CatBoost

# K-En Yakın Komşu (KNN)

---

**Tahminler gözlem benzerliğine göre yapılır.**

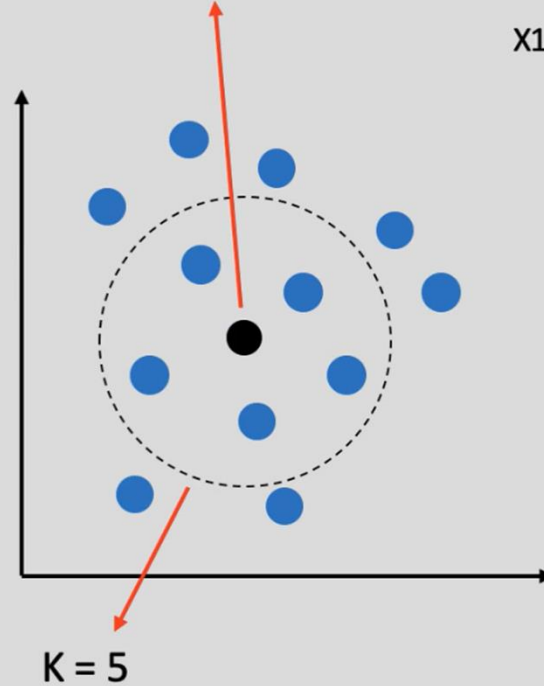
**Bana arkadaşını söyle sana kim olduğunu söyleyeyim.**

- Parametrik olmayan bir yöntemdir
  - Anlaşılması, kullanılması, uygulaması kolaydır
  - Büyük veri setlerinde performans açısından çokta iyi değildir
- Sınıflandırma problemleri için ortaya çıkmış sonra regresyon içinde uyarlanmıştır



# K-En Yakın Komşu

Ortalama Değer



En yakın K adet gözlemin  
y değerlerinin ortalaması alınır.

X1 = 50, X2 = 230

Y tahmini nedir?

Y	X1	X2
100	56	241
120	85	250
.	.	.
.	.	.
140	56	231

$$d(i, j) = \sqrt{\sum_{k=1}^p (x_{ik} - x_{jk})^2}$$

Öklid ya da benzeri bir uzaklık hesabı ile  
her bir gözleme uzaklık hesaplanır.



# K-En Yakın Komşu

## KNN Basamakları

- Komşu sayısını belirle (K)
- Bilinmeyen nokta ile diğer tüm noktalar ile arasındaki uzaklıkları hesapla
- Uzaklıkları sırala ve belirlenen k sayısına göre en yakın olan k gözlemi seç
- Sınıflandırma ise en sık sınıf, regresyon ise ortalama değeri tahmin değeri olarak ver.



# Destek Vektör Regresyonu (SVR)

---

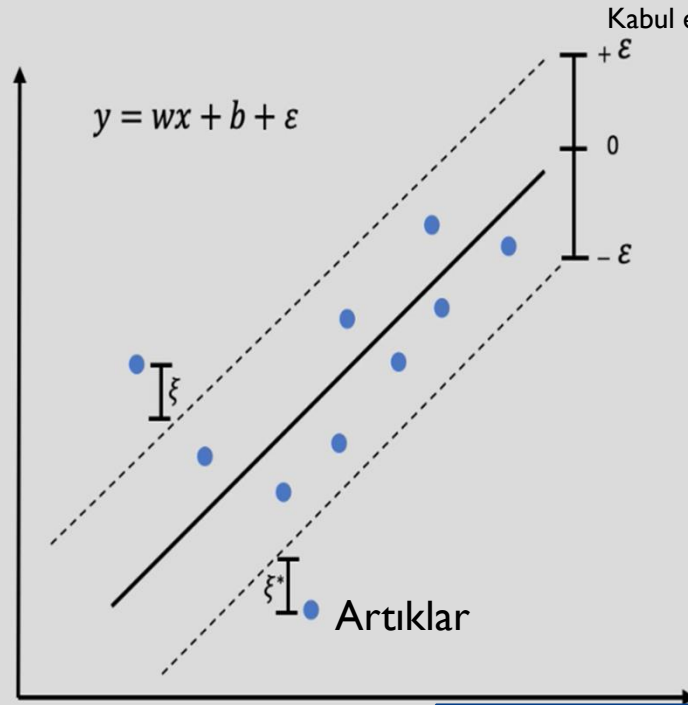
**Amaç, bir marjin aralığına maksimum noktayı en küçük hata ile alabilecek şekilde doğru ya da eğriyi belirlemektir.**

Smola (1996) ve Drucker (1997)





# (Doğrusal) Destek Vektör Regresyonu (SVR)



Minimizasyon Problemi:

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^m (\xi_i + \xi_i^*)$$

C: ceza parametresi

Kısıtlar:

$$y_i - (w * x_i) - b \leq \varepsilon + \xi_i$$

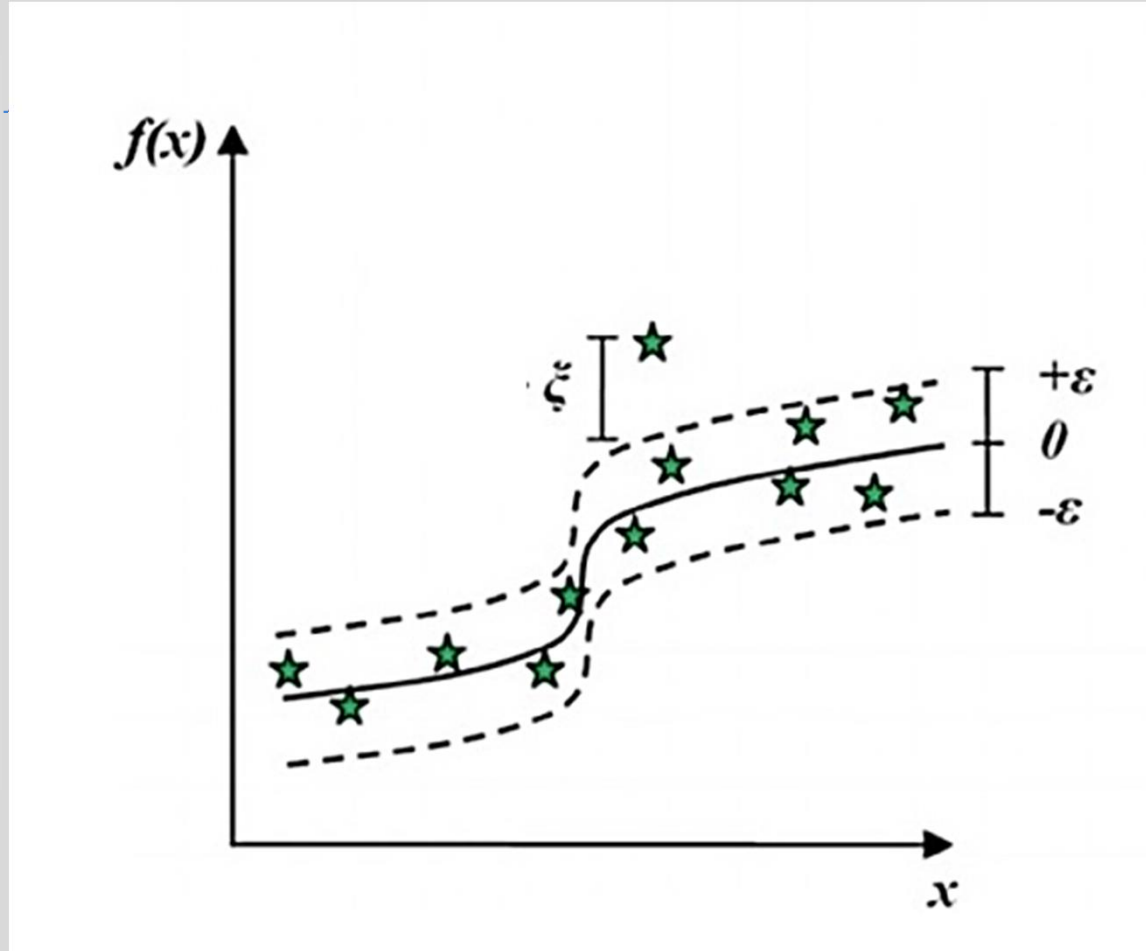
$$(w * x_i) + b - y_i \leq \varepsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \geq 0, \quad i = 1, \dots, m$$

$\xi_i < \varepsilon$  olduğu sürece önemsenmez. Bu artıklar  $\xi_i > \varepsilon$  olduğunda aslında bu artıklar regresyon eğrisini belirliyor. Ama biz C parametresi ile yine de tüm otoriteyi onun eline bırakmıyoruz.

Öyle bir regresyon denklemi bulacağız ki gerçek değerler ile tahmin edilen değerler arasındaki farklar regresyon eğrisinin iki yönünden belirli bir  $\varepsilon + \xi_i$  değerinden daha uzakta olmayacak.

(Doğrusal Olmayan)  
**Destek Vektör  
Regresyonu  
(SVR)**



SVR modeli uygulanırken Radial Basis Function (RBF) metodu ile birlikte uygulandığında doğrusal olmayan bir aralık çizmiş oluruz.





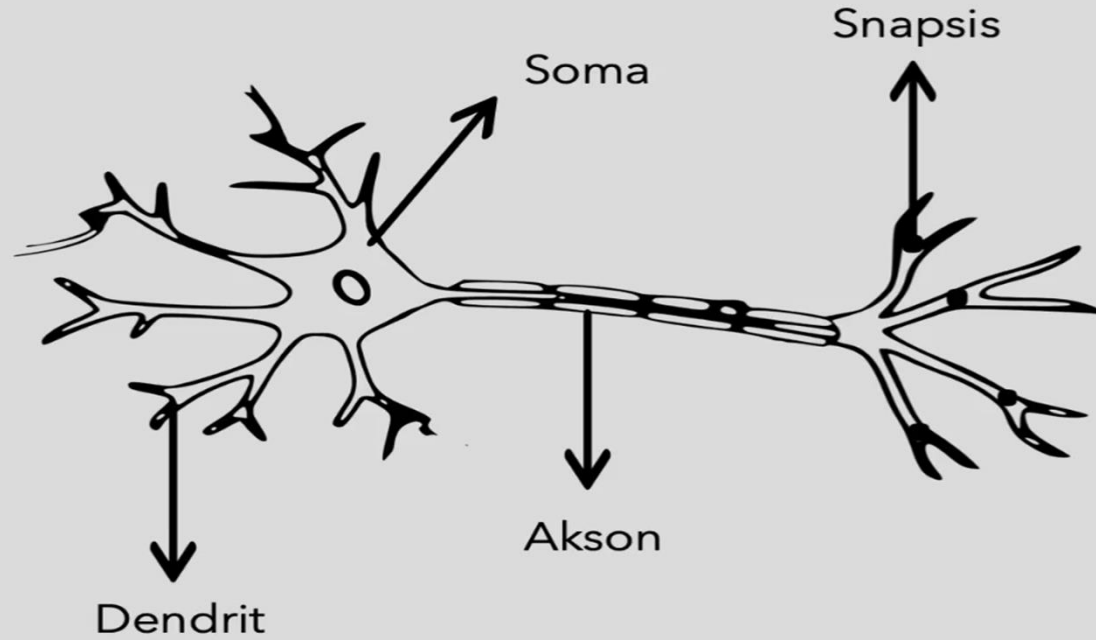
# Yapay Sinir Ağları

---

**İnsan beyninin bilgi işleme şeklini referans alan sınıflandırma ve regresyon problemleri için kullanılabilen kuvvetli makine öğrenmesi algoritmalarından birisidir.**



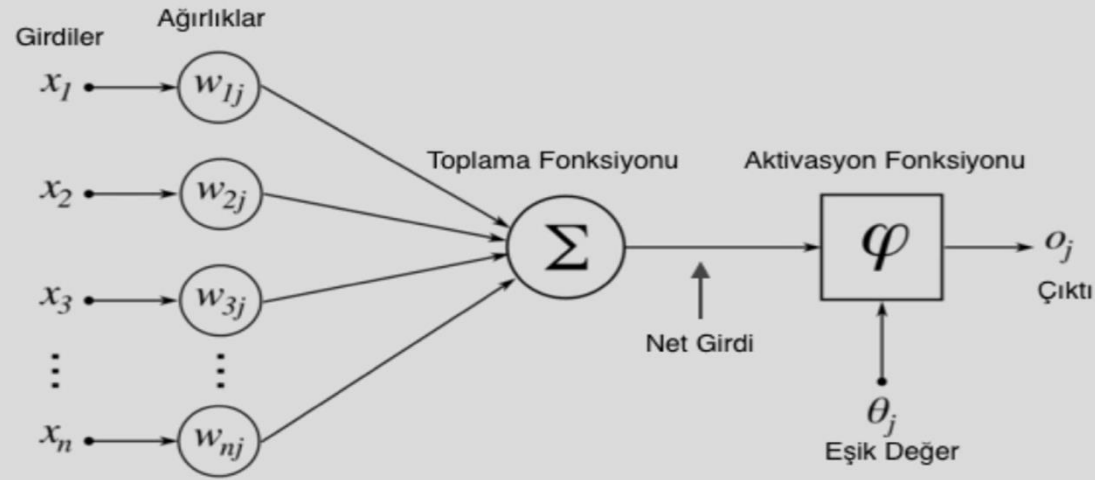
# Sinir Hücresi



- ▶ Dendrit'ler alınan sinyalleri çekirdeğe iletmekle görevlidirler.
- ▶ Hücre çekirdeği ile dentritler arasında bir iletişim gerçekleşmektedir. Dentritlere göre iletişim farklılık göstermekte ve hücre çekirdeği alınan sinyaller konusunda seçicilik yapabilmektedir. Bu durum dentritlerin hücre çekirdeğine ilettiği bilgilerin hücre çekirdeği tarafından **farklı ağırlıklar ile dikkate alınabileceğini ifade eder**.
- ▶ Somaların görevi dentritler tarafından iletilen sinyalleri bir merkezde toplamaktır.
- ▶ Aksonun görevi çekirdekten gelen bilgileri alarak diğer hücrelere iletmektir.
- ▶ Fakat sinyaller diğer hücrelere aktarılmadan önce sinapsislerde bir ön işleme tabi tutulmaktadır. Sinapsislerin görevi gelen sinyalleri belirli bir eşik değerine denk getirecek şekilde değiştirmek, işlemek ve sonrasında diğer hücrelere aktarmaktır.



# Yapay Sinir Hücresi



- ▶ Girdiler = Dentritler.  $x_1, x_2, x_3 \dots x_n \rightarrow$  Bağımsız Değişkenlerin Değerleridir
- ▶ Toplama Fonksiyonu = Soma
- ▶ Dentritler ile soma arasındaki etkileşimde bu etkileşimin ağırlıkları belirleniyor. Yani girdi olarak verilen değişkenlerin değerlerinin çıktıya (bağımlı değişkene) olan etkileri kontrol ediliyor.
- ▶ Aktivasyon Fonksiyonu = Sinapsis

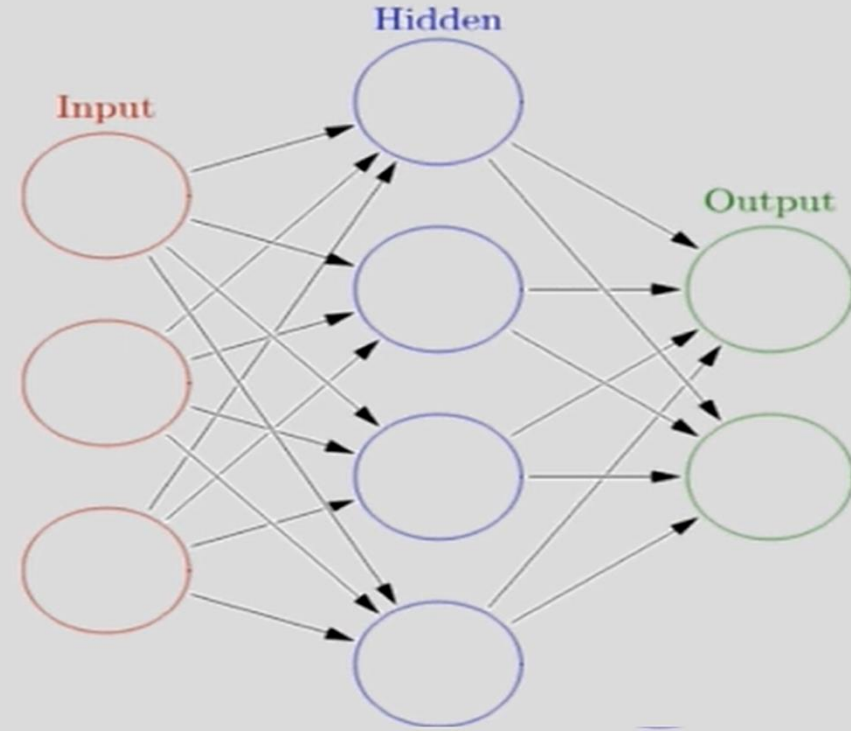
Amacımız hatayı minimum yapacak şekilde bir matematiksel formül veya kural seti oluşturmaktır. Amacımız her zaman aynıdır. Bağımlı değişkeni bulmak üzere bu ağırlıkları öyle bir bulmalıyız ki neticesinde minimum hataya sahip olalım.

Başlangıçta rastgele atanan ağırlık değerleri ile ağ çalışmaya başlıyor ve belirli bir iterasyon sayısınca optimum ağırlıklar bulunmaya çalışılıyor. Her iterasyonda bulunan hatalar (gerçek ve tahmin edilen arasındaki) geriye yayılarak optimum değerler bulunmaya çalışılıyor. (Geriye yayılım Algoritması). Bu ağırlıkların değişmesi aslında öğrenme işlemi olmaktadır.



**Bu basit sinir hücreleri bir araya gelerek yapay sinir ağlarını oluşturur.**

# Yapay Sinir Ağı



- Ağı oluşturan birimlere nöron adı verilir.
- Ağ Input (Girdi), Hidden (Gizli) ve Output (Çıktı) katmanlarından oluşur.
- Birden fazla gizli katman olabilir. Her katmanda birbirinden farklı sayıda nöron bulunabilir.
- Eğitim sırasında yine girdi değerleri ile çıktı değerleri birlikte sunularak elde edilen tahmin değerleri gerçek değerler ile karşılaştırılır. Bunun sonucunda bir epok yapısıyla hatalar geriye doğru yayılarak ağırlıklar ve katmanların optimum değerleri bulunmaya çalışılarak çıktı değeri sınıflandırma veya regresyon problemi için elde edilmeye çalışılır.
- Yapay Sinir Ağlarına, Çok Katmanlı Algılayıcılar ismi de verilir.
- Çok Katmanlı Algılayıcılarda en sık kullanılan algoritma geriye yayılım algoritmasıdır. Bu algoritmada Delta kuralı uygulanır.



# Classification and Regression Trees (CART)

---

**Amaç veri seti içerisindeki karmaşık yapıları basit karar yapılarına dönüştürmektir.**

**Heterojen veri setleri belirlenmiş bir hedef değişkene göre homojen alt gruplara ayrılır.**

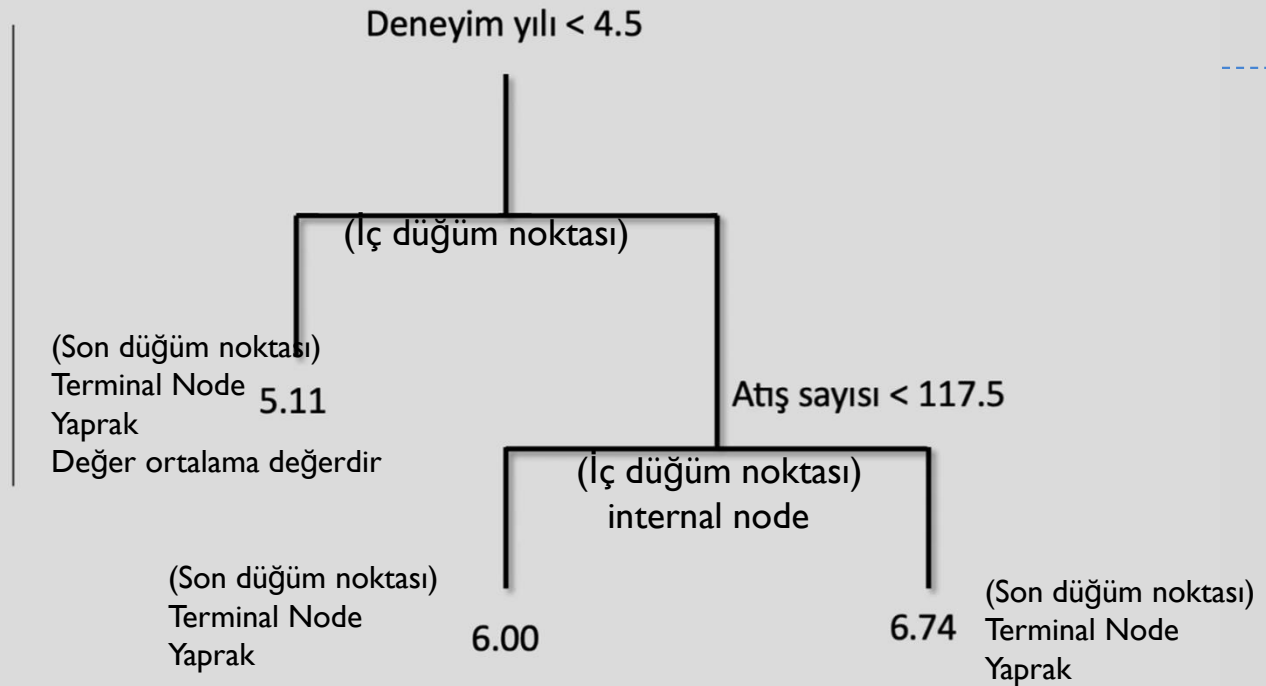
**Bu alt gruplara ayırma işlemi için bazı karar kuralları kullanılır, gini, ki kare, entropy vb.**

Breiman 1984



## Regresyon Problemi Karar Ağacı Yapısı

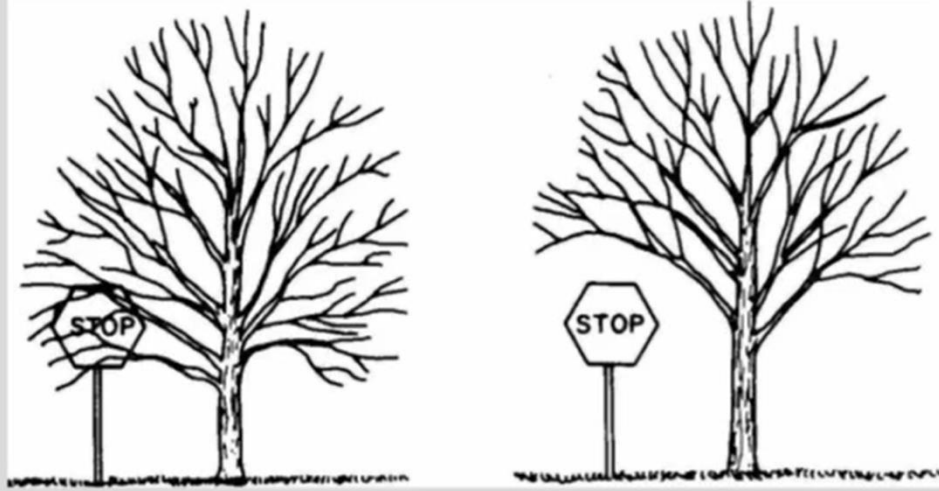
### Regresyon Ağaçları



- Elimizdeki basketbolcuların özellikleri ve maaş bilgisi olan veri setinde deneyim yılı (years) ve atış sayısı (Hit) üzerinden böyle bir karar ağacı oluşturulmuş olsun. Yeni bir veri geldiğinde (yeni bir basketbolcu) maaşının ne olacağını bu karar ağacı kurallarına göre bulabiliriz.
- Açıklayıcı değişkenleri bölgelere ayıran noktalar iç düğüm noktalarıdır.
- En tepedeki değişken hedef değişkeni açıklayan en önemli değişken demektir. Aşağıya doğru değişkenlerin önem düzeyleri azalmaktadır.
- Diğer modellere göre daha az karmaşıktır fakat daha güçsüz bir modeldir.



# Regresyon Ağaçları



Dallanmalar ezberleme gibi bazı problemleri de beraberinde getirir.

Çok fazla dallanma aslında örnek veri setinin çok iyi temsil edilmesini ifade ediyor. Bölebildiğimiz yere kadar bölüp bir nokta geldiğinde eğitim içerisinde çok başarılı bir tahmin gerçekleştirilebilir. Fakat hiç bilmediği test verisine geldiğinden ezberleme probleminden dolayı doğru bir tahmin gerçekleşmiyor. (Aşırı Dallanma)

Bunun çözümü budama işlemidir. Belirlenen karmaşıklık parametresine göre ağaçların dallanması bir yerden kesilip dallanma belirli bir noktada durdurulmuş oluyor. **Bu parametre ceza parametresi olarak da isimlendiriliyor. Burada karmaşıklık parametresinin nasıl bulunacağı söz konusu olacak.**

Genellenebilirlik kaygısı olmadığında başarılı bir model kurulmuşsa bu regresyon ağaçları kullanılabilir. Örneğin bir şirketteki maaş hesaplama gibi.

Fakat genelleme yapılacaksa bu model bir şehre bir ülkeye açılacaksa bu durumda regresyon ağaçları zayıf kalacaktır.



# Bagging

---

**Temeli bootstrap yöntemi ile oluşturulan birden fazla karar ağacının ürettiği tahminlerin bir araya getirilerek değerlendirilmesine dayanır.**

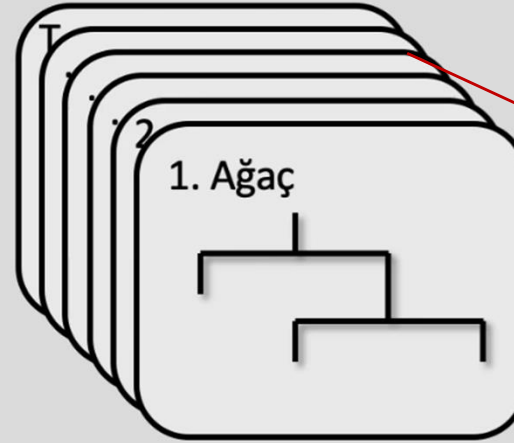


# Bagging Çalışma Prensibi

1,2,3, ... , m

m: gözlem sayısı

↓  
T adet ağaç için  
n'er adet gözlem ( $n < m$ )  
bootstrap yöntemi ile seçilir.



Rastgele örnekleme demektir



T adet karar ağacı modelinin ürettiği T adet tahmin değerini bir araya getir.

Ağaçların birbirlerine bağımlılıkları yoktur

Çekilen örneklerin 2/3'ü ağaçların oluşturulması geriye kalanları test edilmesi için kullanılır.

Varyansı düşüren ve ezberlemeye karşı dayanıklı olan bir yöntemdir.





# Random Forests

---

**Temeli birden çok karar ağacının ürettiği tahminlerin bir araya getirilerek değerlendirilmesine dayanır.**



# Random Forests

- Bagging (Breiman, 1996) ile Random Subspace (Ho, 1998) yöntemlerinin birleşimi ile oluşmuştur.
  - Ağaçlar için gözlemler bootstrap rastgele örnek seçim yöntemi ile **değişkenler random subspace yöntemi ile seçilir.**
  - Karar ağacının her bir düğümünde en iyi dallara ayırıcı (bilgi kazancı) değişken tüm değişkenler arasında rastgele seçilen daha az sayıdaki değişken arasından seçilir.
  - Ağaç oluşturmada veri setinin 2/3'ü kullanılır. Dışarıda kalan veri ağaçların performans değerlendirmesi ve değişken öneminin belirlenmesi için kullanılır.
  - Her düğüm noktasında rastgele değişken seçimi yapılır. (regresyon'da  $p/3$ , sınıflama'da  $\sqrt{p}$ )
- Nihai tahmin için ağaçlardan tahmin değerleri talep edilirken her bir ağacın daha önce hesaplanan hata oranları göz önüne alınarak ağaçlara ağırlık verilir.



---

# TEŞEKKÜRLER

