# Makine Öğrenmesi

## Sınıflandırma ve Ağaçlar

**İlker Birbil ve Utku Karaca**

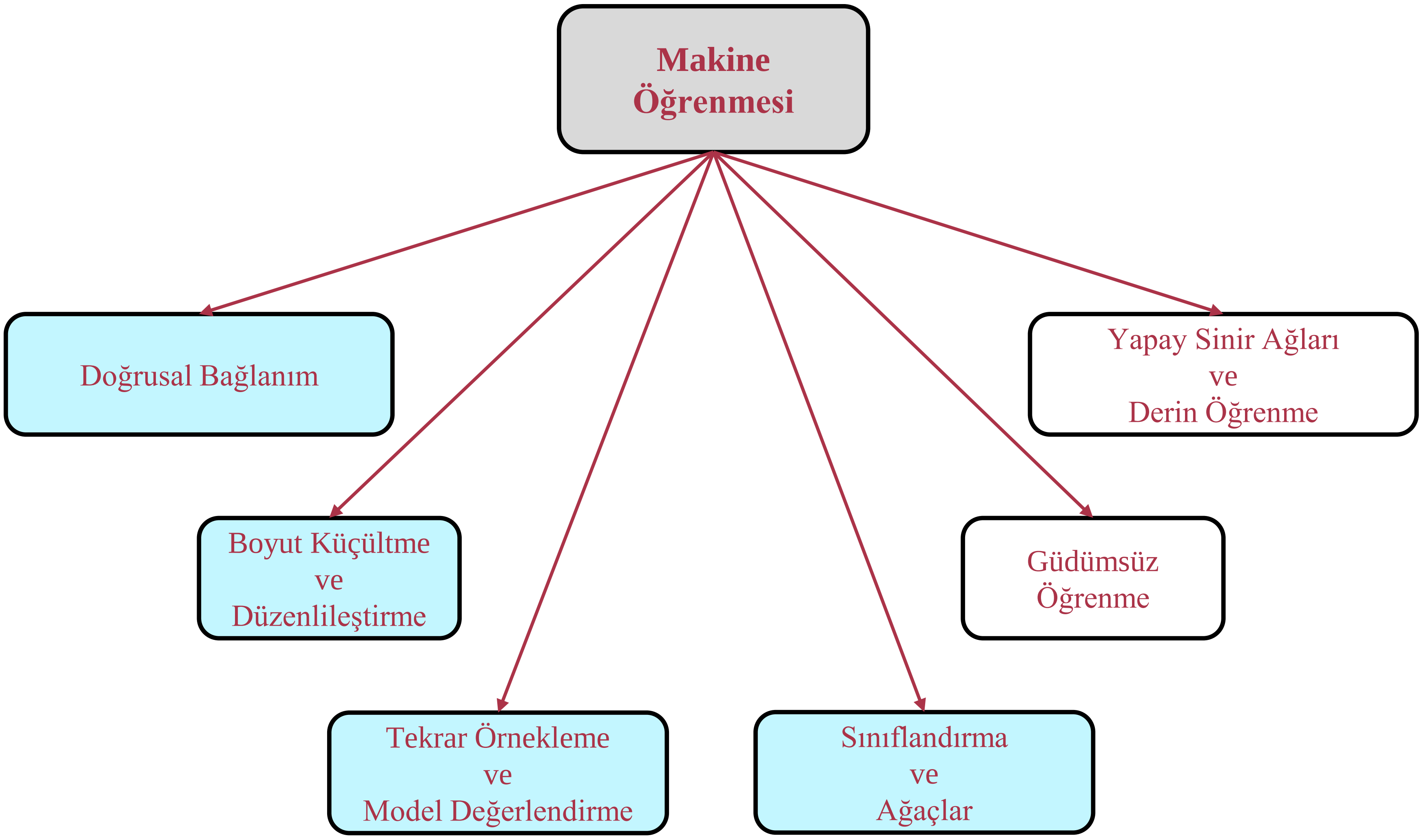
Erasmus Üniversitesi Rotterdam

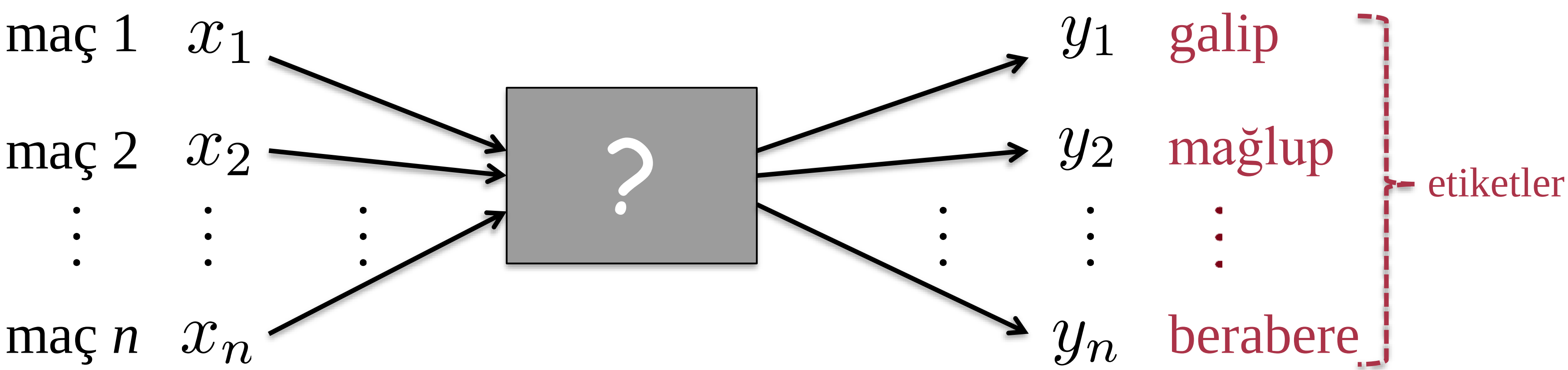
İstanbul'da Makine Öğrenmesi

27 Ocak – 2 Şubat, 2020



EUR
D4C
Bol Bilim
Veri Defteri
@sibirbil


Makine Öğrenmesi
Doğrusal Bağlanım
Boyut Küçültme ve Düzenlileştirme
Tekrar Örnekleme ve Model Değerlendirme
Sınıflandırma ve Ağaçlar
Güdümsüz Öğrenme
Yapay Sinir Ağları ve Derin Öğrenme




eğitim verisi

$$\{(x_i, y_i) : 1, \ldots, n\}$$

# Lojistik Bağlanım (Logistic Regression)

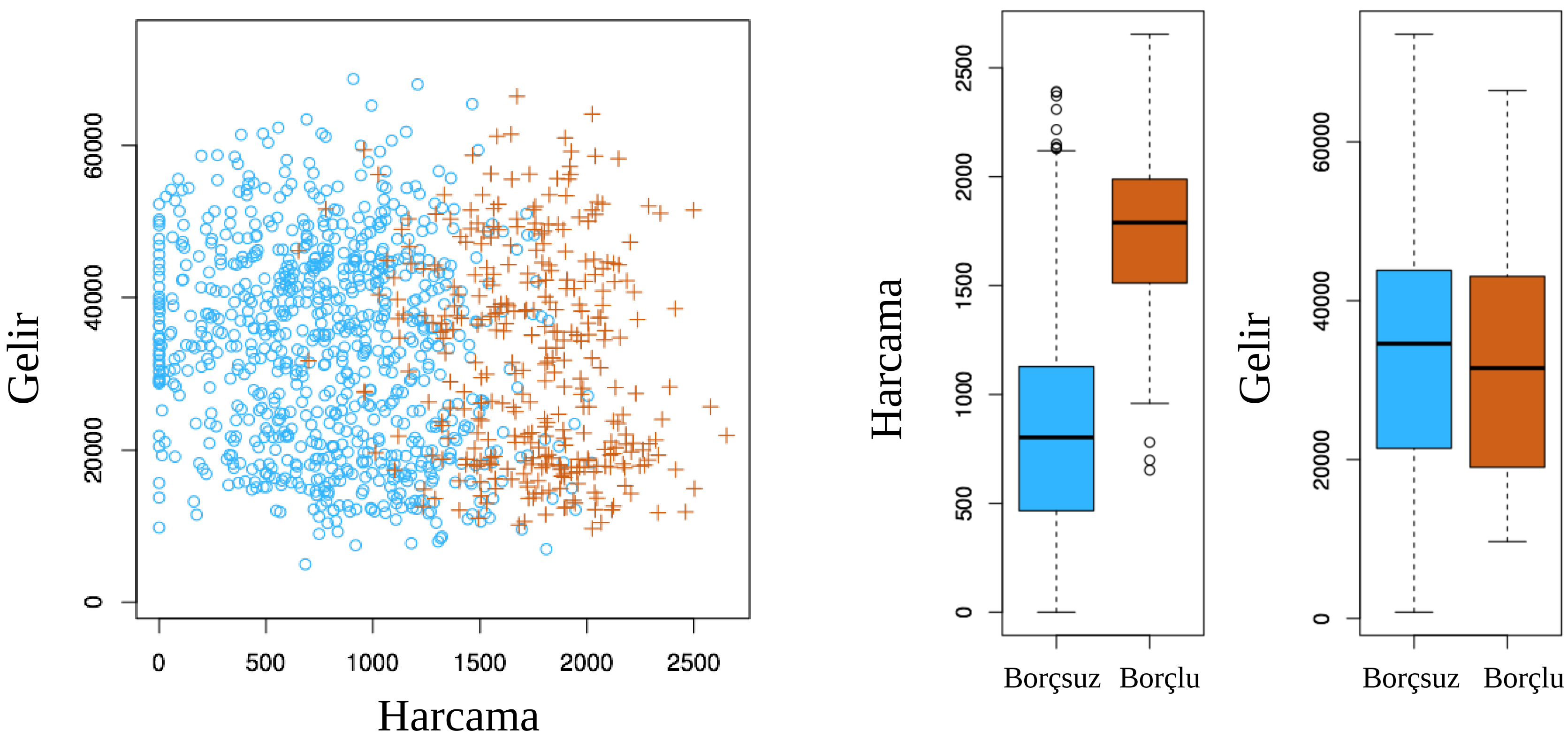
$X_1$ : toplam harcama (bakiye)
$X_2$ : yıllık gelir
$X_3$ : öğrenci (1), öğrenci değil (0)

$$Y = \begin{cases} \text{kart borcu kalan (borçlu)}, & 1; \\ \text{kart borcu kalmayan (borçsuz)}, & 0. \end{cases}$$

$$p(X) = \mathbb{P}(Y = 1|X) \quad ?$$

$$p(X) = \mathbb{P}(Y = 1|X) > \tau$$

eşik değer (threshold)



Gelir
Harcama
0
500
1000
1500
2000
2500
20000
40000
60000
Borçsuz
Borçlu

$$p(X) = \mathbb{P}(Y = 1|X) \quad ?$$

$$\boxed{p = 1}$$

$$p(X) = \beta_0 + \beta_1 X \longrightarrow (-\infty, +\infty)$$

$$p(X) = \sigma(\beta_0 + \beta_1 X) \longrightarrow [0, 1]$$

?

Sigmoid Fonksiyonu

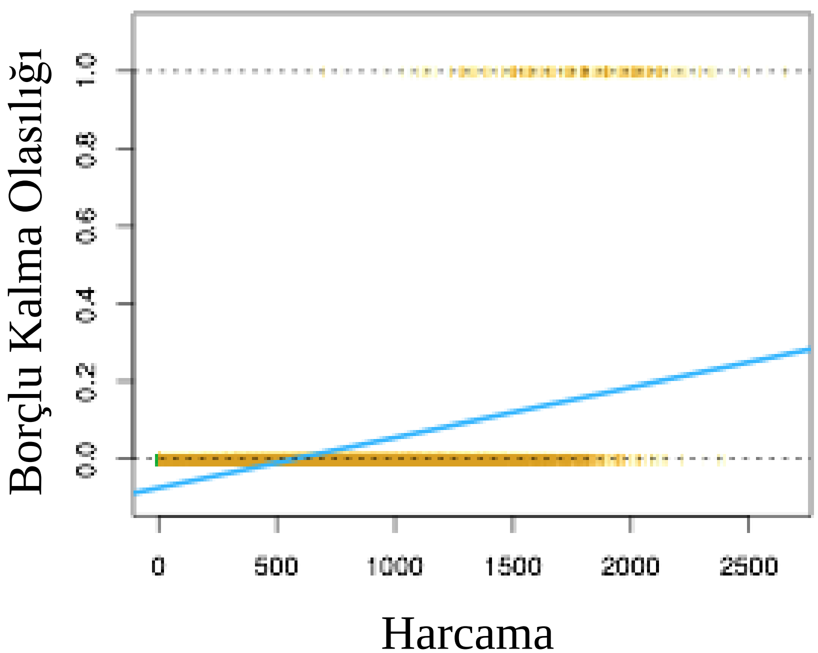
$$\sigma(y) = \frac{e^y}{1 + e^y}$$

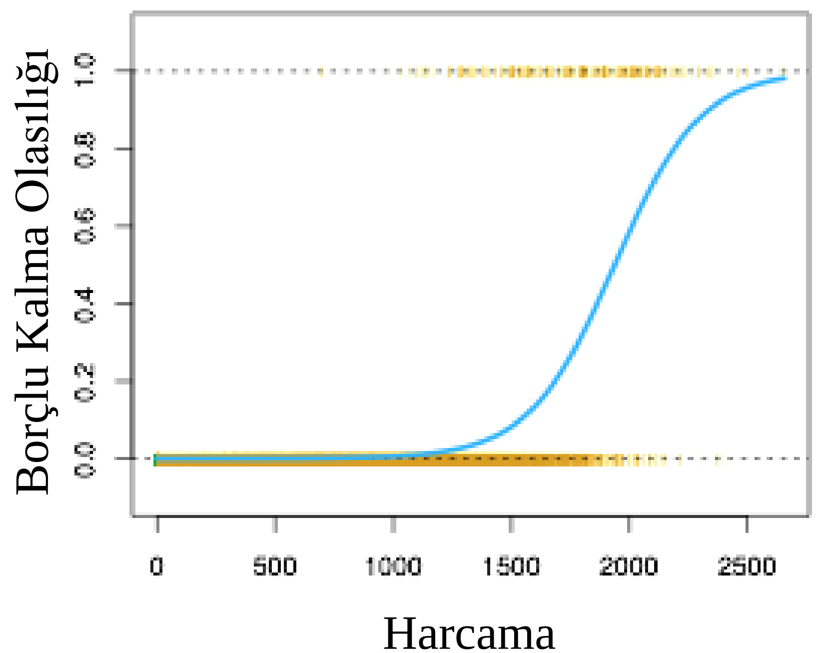
Lojistik Fonksiyonu

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}} \longrightarrow [0, 1]$$

$$\boxed{p(X) = \mathbb{P}(Y = 1|X) \quad ?}$$

$$p(X) = \beta_0 + \beta_1 X \longrightarrow (-\infty, +\infty) \qquad p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}} \longrightarrow [0, 1]$$




$$\log\left(\underbrace{\frac{p(X)}{1 - p(X)}}_{\text{göreceli olasılıklar (odds)}}\right) = \beta_0 + \beta_1 X$$

$\boxed{p > 1}$

$$p(X) = \sigma(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p)$$

Lojistik Fonksiyonu

$$p(X) = \frac{e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}$$

$$\beta_0, \beta_1, \ldots, \beta_p?$$

$$\downarrow$$

eğitim verisi

$$\{(x_i, y_i) : 1, \ldots, n\}$$

$$\overbrace{1-\mathbb{P}(Y=1|X=x_2)}^{1-p(x_2)} \qquad \overbrace{1-\mathbb{P}(Y=1|X=x_5)}^{1-p(x_5)}$$

$$\boxed{(x_1,1)}, \quad \boxed{(x_2,0)}, \quad \boxed{(x_3,1)}, \quad \boxed{(x_4,1)}, \quad \boxed{(x_5,0)}$$

$$\underbrace{\mathbb{P}(Y=1|X=x_1)}_{p(x_1)} \qquad \underbrace{\mathbb{P}(Y=1|X=x_3)}_{p(x_3)} \qquad \underbrace{\mathbb{P}(Y=1|X=x_4)}_{p(x_4)}$$

$$p(x_1)p(x_3)p(x_4)(1-p(x_2))(1-p(x_5))$$

Olabilirlik (Likelihood) Fonksiyonu

$$\ell(\beta_0,\ldots,\beta_p) = \prod_{i:y_i=1} p(x_i) \prod_{j:y_j=0} (1-p(x_j))$$

# En Büyük (Maximum) Olabilirlik

$$\boxed{\ell(\beta_0, \ldots, \beta_p) = \prod_{i:y_i=1} p(x_i) \prod_{j:y_j=0} (1 - p(x_j))}$$

$$\max_{\beta_0,\ldots,\beta_p} \ell(\beta_0, \ldots, \beta_p)$$

$$\downarrow$$

$$\max_{\beta_0,\ldots,\beta_p} \log(\ell(\beta_0, \ldots, \beta_p))$$

$$\downarrow$$

$$\max_{\beta_0,\ldots,\beta_p} \sum_{i:y_i=1} \log p(x_i) + \sum_{j:y_j=0} \log(1 - p(x_j))$$

$$\downarrow$$

İçbükey Fonksiyon

$$\downarrow$$

$$\hat{\beta}_0, \ldots, \hat{\beta}_p$$

$p = 1$

| | **Katsayı** | **St. Hata** | ***Z* istatistiği** | ***p* değeri** |
|---|---|---|---|---|
| **Kesme nok.** | -3,5041 | 0,0707 | -49,55 | < 0,0001 |
| **Öğrenci (1)** | 0,4049 | 0,1150 | 3,52 | 0,0004 |

$p > 1$

| | **Katsayı** | **St. Hata** | ***Z* istatistiği** | ***p* değeri** |
|---|---|---|---|---|
| **Kesme nok.** | -10,8690 | 0,4923 | -22,08 | < 0,0001 |
| **Harcama** | 0,0057 | 0,0002 | 24,74 | < 0,0001 |
| **Gelir** | 0,0030 | 0,0082 | 0,37 | 0,7115 |
| **Öğrenci (1)** | -0,6468 | 0,2362 | -2,74 | 0,0062 |

Parazit Etkisi (Confounding Effect)

# Tahmin Yapma

| | **Katsayı** | **St. Hata** | ***Z* istatistiği** | ***p* değeri** |
|---|---|---|---|---|
| **Kesme nok.** | -10,8690 | 0,4923 | -22,08 | < 0,0001 |
| **Harcama** | 0,0057 | 0,0002 | 24,74 | < 0,0001 |
| **Gelir** | 0,0030 | 0,0082 | 0,37 | 0,7115 |
| **Öğrenci (1)** | -0,6468 | 0,2362 | -2,74 | 0,0062 |

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}$$

$$X = \begin{bmatrix} 1.500 \\ 40.000 \\ 1 \end{bmatrix}$$

$$\hat{p}(X) = \frac{e^{-10,869+0,0057\times 1.500+0.003\times 40-0.6468\times 1}}{1+e^{-10,869+0,0057\times 1.500+0.003\times 40-0.6468\times 1}} = 0.058$$

# Doğrusal Ayrımlayıcı Çözümleme
# (Linear Discriminant Analysis - LDA)

$$\mathbb{P}(Y=k|X=x) \quad \leftrightsquigarrow \quad \mathbb{P}(X=x|Y=k)$$

Bayes Kuramı

$$\mathbb{P}(Y=k|X=x) = \frac{\overbrace{\mathbb{P}(Y=k)}^{\pi_k}\overbrace{\mathbb{P}(X=x|Y=k)}^{f_k(x)}}{\sum_{l=1}^{K}\mathbb{P}(Y=l)\mathbb{P}(X=x|Y=l)}$$

Ardıl Olasılık (Posterior Probability)

$$p_k(x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K}\pi_l f_l(x)}$$

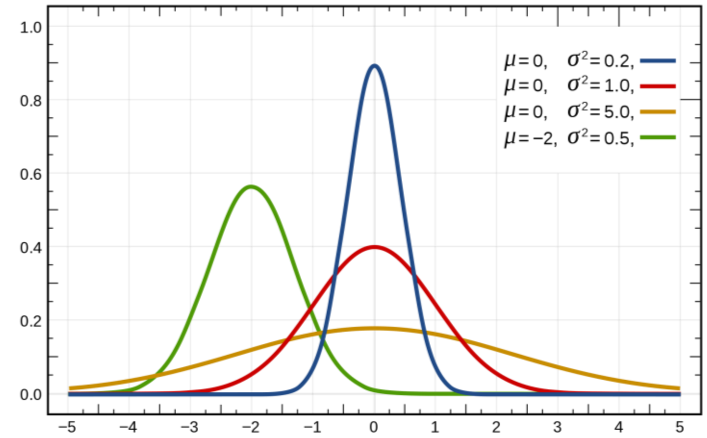
$\pi_k$ ? $\qquad\qquad$ $f_k(x)$ ?

$$f_k(x) \text{ ?}$$

$$p = 1$$

$$p_k(x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^K \pi_l f_l(x)}$$

**Kabul 1:** $X \sim N(\mu, \sigma)$




$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right)$$

**Kabul 2:** $\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_K^2 = \sigma^2$

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_k)^2\right)}{\sum_{l=1}^K \pi_l \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_l)^2\right)}$$

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_k)^2\right)}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_l)^2\right)}$$

$$\begin{aligned}
\log(p_k(x)) &= \log\left(\frac{\pi_k \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_k)^2\right)}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_l)^2\right)}\right) \\
&\vdots \\
&= \underbrace{x\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)}_{\delta_k(x)} - C
\end{aligned}$$

*k*’den bağımsız
sabit terim

$$\delta_k(x) = x\,\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

$$\hat{\mu}_k = \frac{1}{n_k}\sum_{i:y_i=k} x_i \qquad \hat{\sigma}^2 = \frac{1}{n-K}\sum_{k=1}^{K}\sum_{i:y_i=k}(x_i - \hat{\mu}_k)^2 \qquad \hat{\pi}_k = \frac{n_k}{n}$$

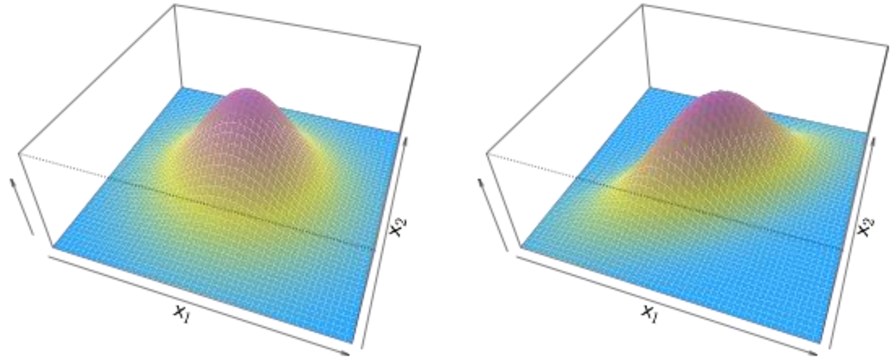
$$\boxed{\hat{\delta}_k(x) = x\,\frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k)}$$

$$\boxed{X = x} \qquad \hat{\delta}_k(x) \qquad \boxed{Y = k}$$

$\boxed{p > 1}$ $\quad X = (X_1, X_2, \ldots, X_p)$

**Kabul 1:** $X \sim N(\mu, \Sigma)$

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_k)^\intercal \Sigma_k^{-1}(x-\mu_k)\right)$$




**Kabul 2:** $\Sigma_1 = \Sigma_2 = \cdots = \Sigma_K = \Sigma$

$$\delta_k(x) = x^{\mathsf{T}}\Sigma^{-1}\mu_k - \tfrac{1}{2}\mu_k^{\mathsf{T}}\Sigma^{-1}\mu_k + \log(\pi_k)$$

$$\hat{\mu}_k = \frac{1}{n_k}\sum_{i:y_i=k} x_i \qquad\qquad \hat{\pi}_k = \frac{n_k}{n}$$

$$\hat{\Sigma} = \frac{1}{n-K}\sum_{k=1}^{K}\sum_{i:y_i=k}(x_i-\hat{\mu}_k)(x_i-\hat{\mu}_k)^{\mathsf{T}}$$

$$\boxed{\hat{\delta}_k(x) = x^{\mathsf{T}}\hat{\Sigma}^{-1}\hat{\mu}_k - \tfrac{1}{2}\hat{\mu}_k^{\mathsf{T}}\hat{\Sigma}^{-1}\hat{\mu}_k + \log(\hat{\pi}_k)}$$

$\boxed{X = x}$ $\qquad$ $\hat{\delta}_k(x) \uparrow\uparrow$ $\qquad$ $\boxed{Y = k}$

$$p(X) = \mathbb{P}(Y = 1|X) > \tau$$

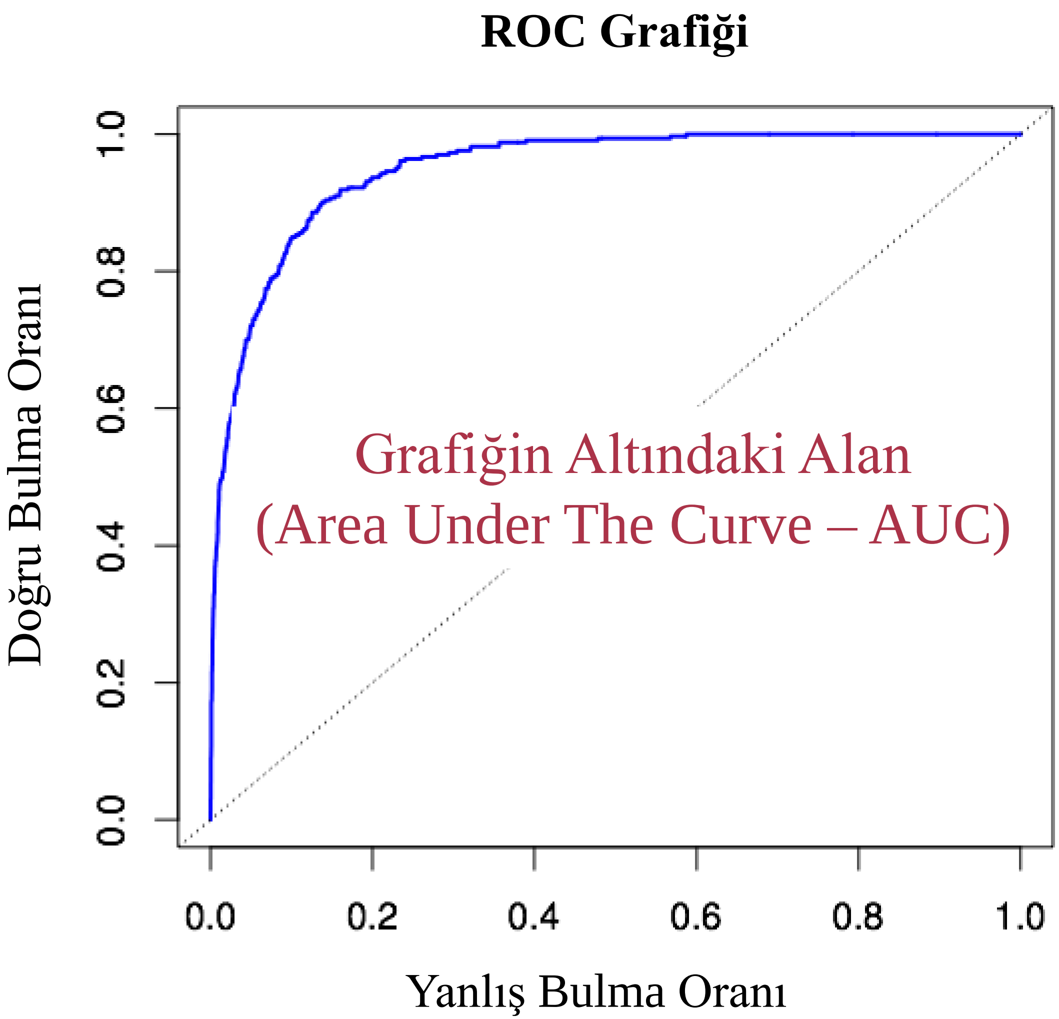
$$\tau = 0.5$$

| | | **Gerçek** | | |
|---|---|---|---|---|
| | | **Borçsuz** | **Borçlu** | **Toplam** |
| **Tahmin** | **Borçsuz** | 9.644 | 252 | 9.896 |
| | **Borçlu** | 23 | 81 | 104 |
| | **Toplam** | 9.667 | 333 | 10.000 |

$$\tau = 0.2$$

| | | **Gerçek** | | |
|---|---|---|---|---|
| | | **Borçsuz** | **Borçlu** | **Toplam** |
| **Tahmin** | **Borçsuz** | 9.432 | 138 | 9.570 |
| | **Borçlu** | 235 | 195 | 430 |
| | **Toplam** | 9.667 | 333 | 10.000 |

# ROC* Grafiği
# (Karar Değerlendirme Grafiği)




* Receiver Operating Characteristics

# Hata Matrisi (Confusion Matrix)

| | | Tahmin Edilen Sınıf | | |
|---|---|---|---|---|
| | | **-** | **+** | **Toplam** |
| **Gerçek Sınıf** | **-** | Gerçek Negatif (GN) | Yanlış Pozitif (YP) | N |
| | **+** | Yanlış Negatif (YN) | Gerçek Pozitif (GP) | P |
| | **Toplam** | N* | P* | |

| | | Tahmin Edilen Sınıf | | |
|---|---|---|---|---|
| | | - | + | Toplam |
| Gerçek Sınıf | - | GN | YP | N |
| | + | YN | GP | P |
| | Toplam | N* | P* | |

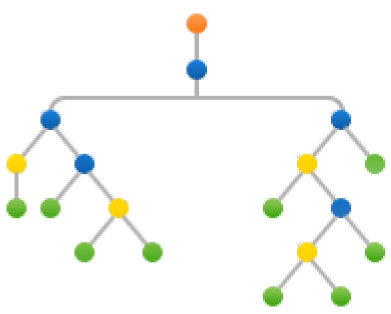
| İsim | Hesap | Eş Anlam |
|---|---|---|
| **Yanlış Bulma Oranı**[a] | YP/N | birinci tip hata (type I error),<br>1 – özgüllük (specifity) |
| **Doğru Bulma Oranı**[b] | GP/P | 1 – ikinci tip hata (type II error),<br>üs, duyarlılık (sensitivity),<br>doğruluk (recall) |
| **Pozitif Tahmin Değeri**[c] | GP/P* | kesinlik (precision),<br>1 – yanlış keşif oranı (false discovery proportion) |
| **Negatif Tahmin Değeri**[d] | GN/N* | |

[a] False Positive Rate
[b] True Positive Rate
[c] Positive Predictive Value
[d] Negative Predictive Value

# Karar Ağaçları (Decision Trees)

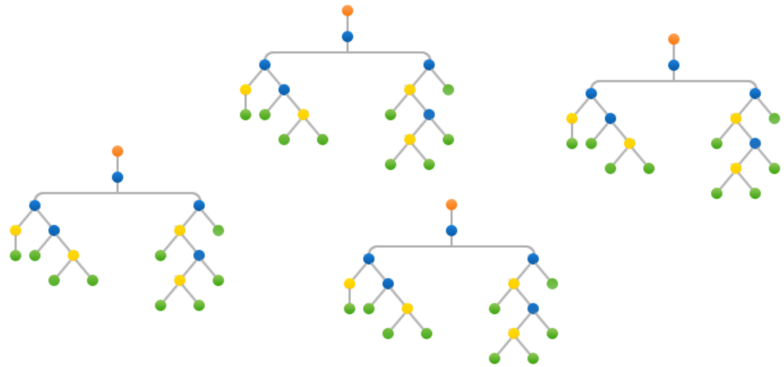
**Amaç:** Değişkenler uzayını basit alt bölgelere ayırmak

Bağlanım Ağaçları
Sınıflandırma Ağaçları

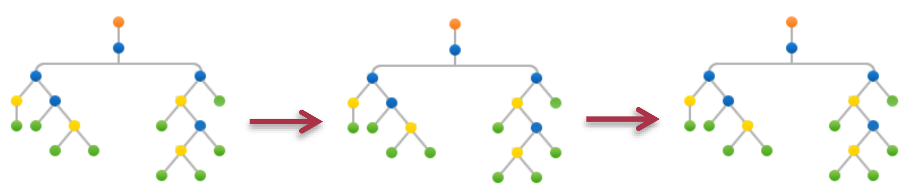


**Topluluk Yöntemleri**

Torbalama (Bagging) ve
Rasgele Ormanlar (Random Forests)

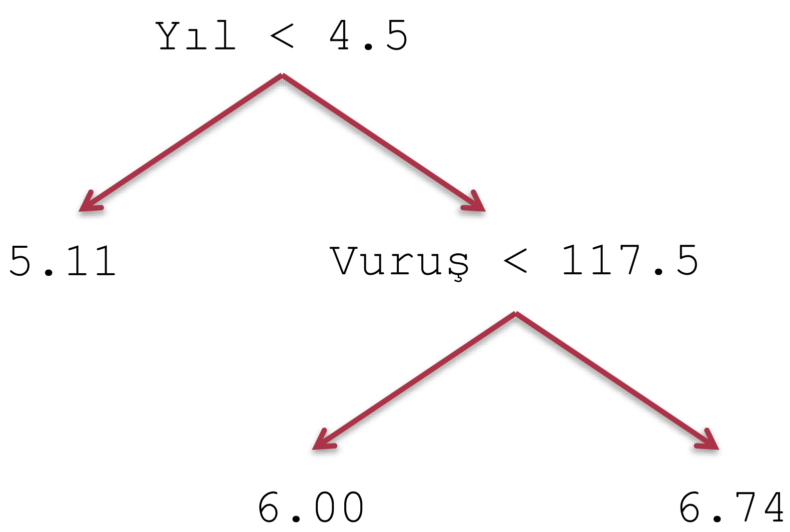


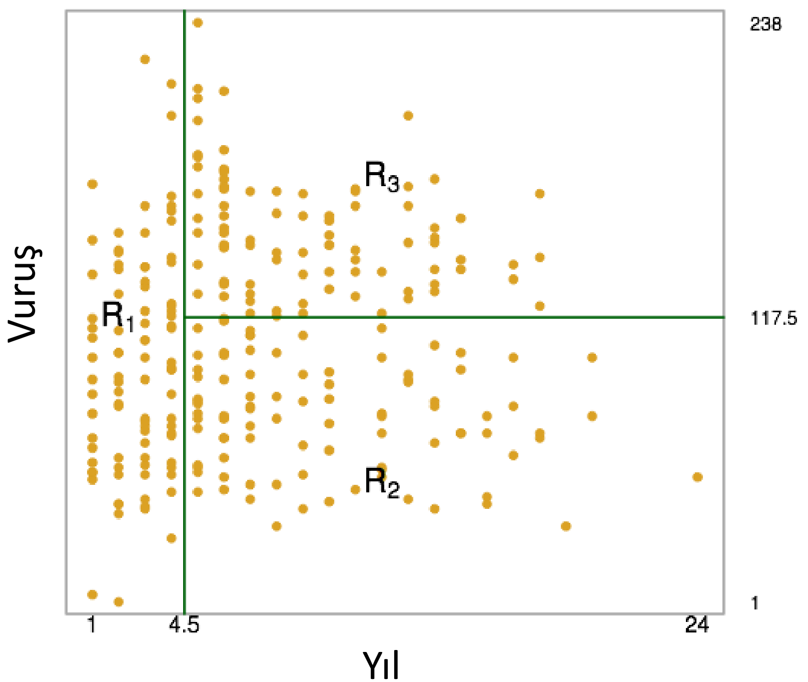
Takviye (Boosting)



# Örnek Ağaç

Bir beyzbol oyuncusunun maaşının yaptığı vuruş ve oynadığı toplam yıla göre tahmin edilmesi



# Bağlanım Ağaçları

$$X_1, X_2, \ldots, X_p \xrightarrow{\text{Çakışmayan bölgeler}} R_1, R_2, \ldots, R_J$$

**Tahmin:** $R_j$ bölgesindeki eğitim verisinin çıktı değerlerinin ($y_i$) ortalaması

$R_1, R_2, \ldots, R_J$ **?**

**Amaç:** KKT değerinin en küçük olduğu bölgelerin bulunması

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$

$\hat{y}_{R_j}$ : $R_j$ bölgesindeki çıktıların ortalaması

# Bağlanım Ağaçları

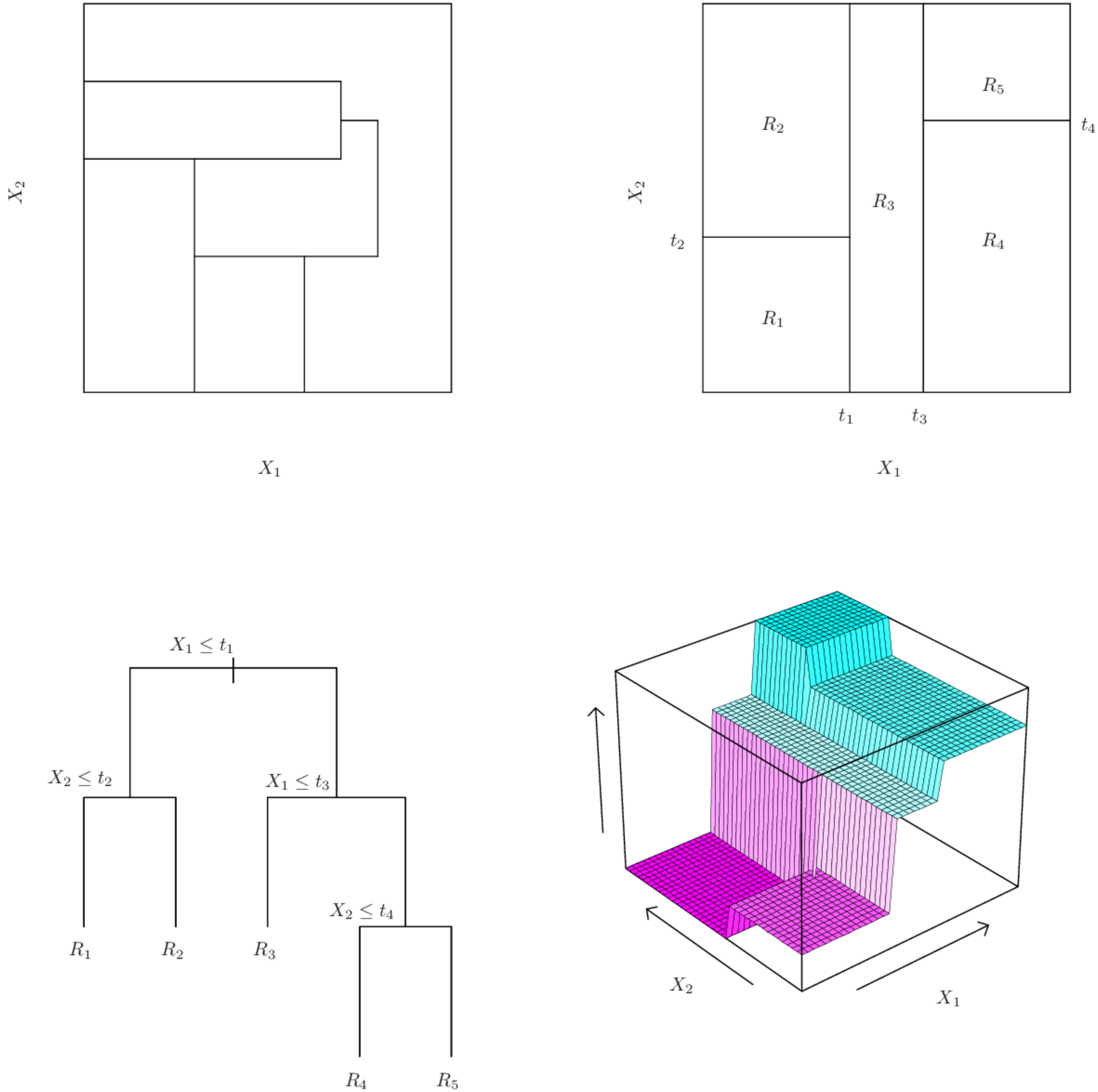
## Özyinelemeli İkili Ayırma (Recursive Binary Splitting)

$$R_1(j,s) = \{X | X_j < s\} \qquad R_2(j,s) = \{X | X_j \geq s\}$$

Aşağıdaki ifadeyi en küçükleyen *j* ve *s* değerlerini bul

$$\sum_{i:x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2$$

Terminal düğümde çok az veri noktası kalınca **dur!**



$X_2$
$X_1$
$R_2$
$R_5$
$t_4$
$R_3$
$R_4$
$t_2$
$R_1$
$t_1$
$t_3$
$X_2$
$X_1$
$X_1 \leq t_1$
$X_2 \leq t_2$
$X_1 \leq t_3$
$X_2 \leq t_4$
$R_1$
$R_2$
$R_3$
$R_4$
$R_5$
$X_2$
$X_1$

# Bağlanım Ağaçları

## Ağaç Budama (Tree Pruning)

**Amaç:** Ağacın tamamı ya da büyük kısmı oluşunca ortaya çıkan aşırı öğrenmenin önüne geçmek (düşük *test* hatası)

$$\sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha|T|$$

$T \subset T_0$ : alt ağaç (subtree)

$|T|$ : $T$ ağacındaki terminal düğüm sayısı

$R_m$ : $m$. terminal düğüme karşılık gelen bölge

$\alpha$ : sabit parametre

$\alpha \uparrow \qquad |T| \downarrow$

$\alpha$ parametresini bulmak için $k$-katlı çapraz geçerlilik sınaması yapılır

# Sınıflandırma Ağaçları

Bağlanım ağaçlarına çok benzer şekilde ilerlenir ancak alt bölgenin *saflık* derecesine bağlı bir hata ölçüsü kullanılır

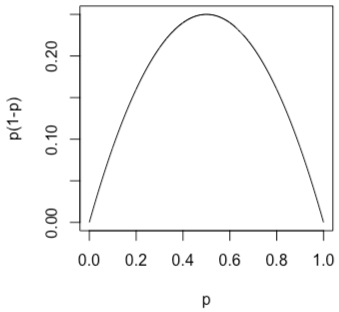
$\hat{p}_{mk}$ : *m*. bölgedeki eğitim verisindeki *k*. sınıftan olanların oranı

**Sınıflandırma Hata Oranı**

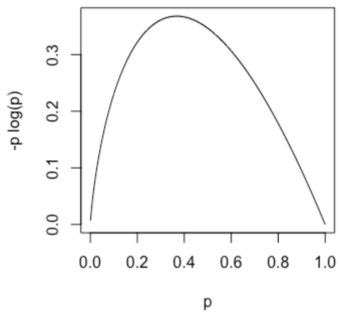
$$E = 1 - \max_k \{\hat{p}_{mk}\}$$

**Gini İndeksi**

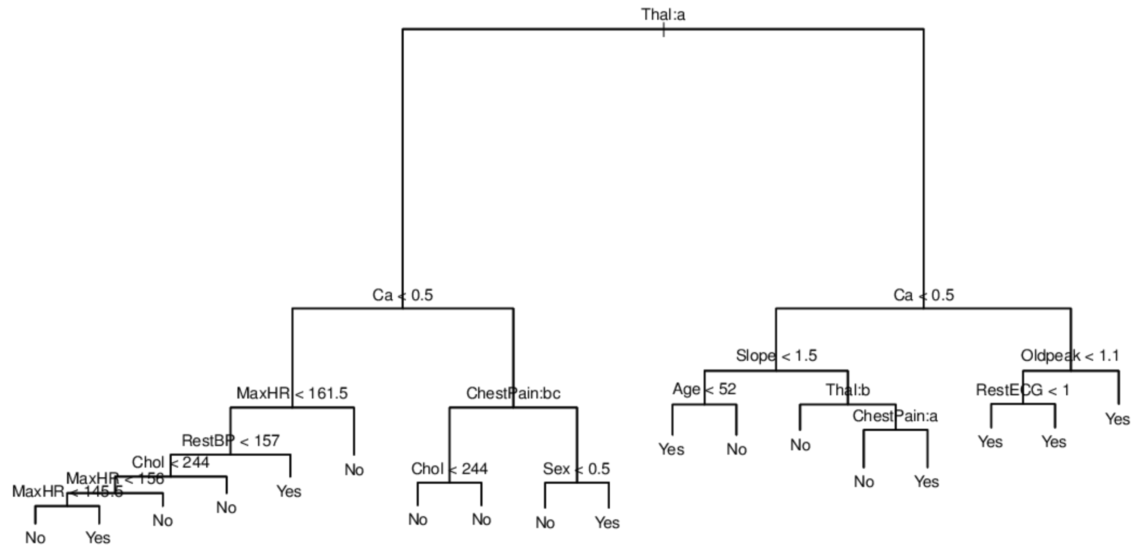
$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

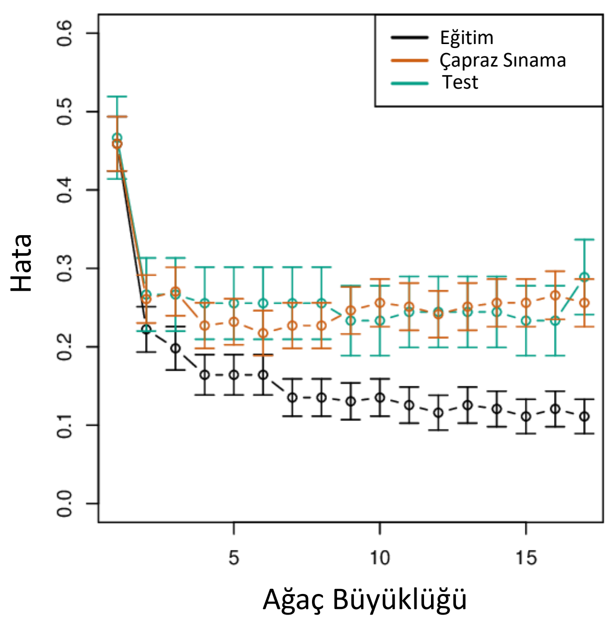


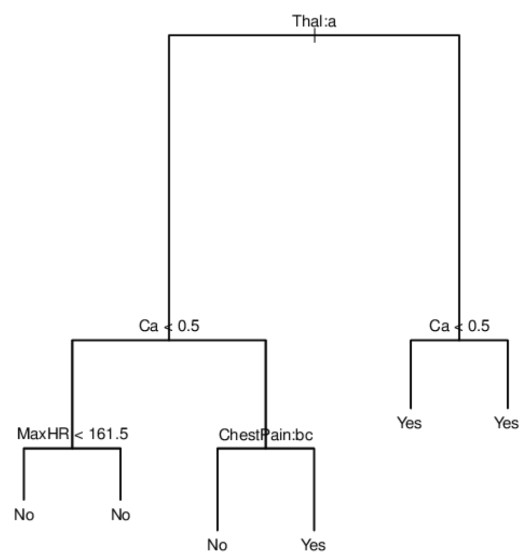

**Entropi**




$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$$


Thal:a
Ca < 0.5
Ca < 0.5
MaxHR < 161.5
ChestPain:bc
Slope < 1.5
Oldpeak < 1.1
Age < 52
Thal:b
ChestPain:a
RestECG < 1
RestBP < 157
Chol < 244
MaxHR < 156
MaxHR < 145.5
Chol < 244
Sex < 0.5
No
Yes
No
No
Yes
No
No
No
No
Yes
Yes
No
No
No
Yes
Yes
Yes
Yes

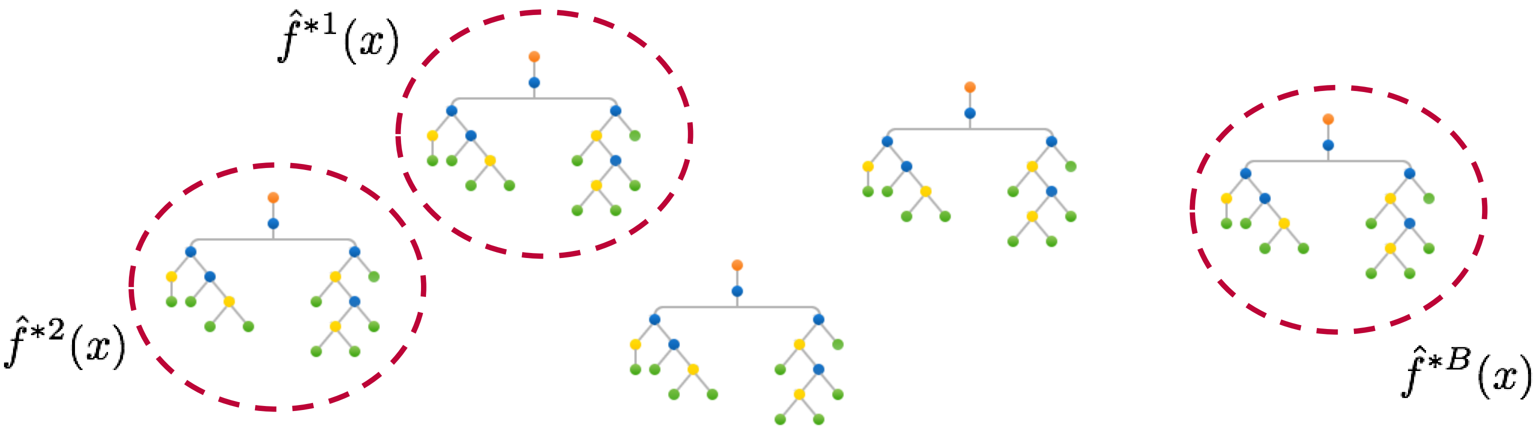

Eğitim
Çapraz Sınama
Test
Hata
Ağaç Büyüklüğü
0.0
0.1
0.2
0.3
0.4
0.5
0.6
5
10
15


Thal:a
Ca < 0.5
Ca < 0.5
MaxHR < 161.5
ChestPain:bc
Yes
Yes
No
No
No
Yes

# Topluluk Yöntemleri

## Torbalama

**Amaç:** Varyansı düşürmek için zorlama tekniğini kullanarak birkaç tane büyük ağaç oluşturulur ve onların tahminlerinin ortalaması (bağlanım) ya da çoğunlukta olan sınıf (sınıflandırma) hesaplanır.




## Bağlanım

$B$ : farklı eğitim kümesi sayısı

$\hat{f}^{*b}(x)$ : $b$. eğitim kümesi ile elde edilen tahmin

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(x)$$

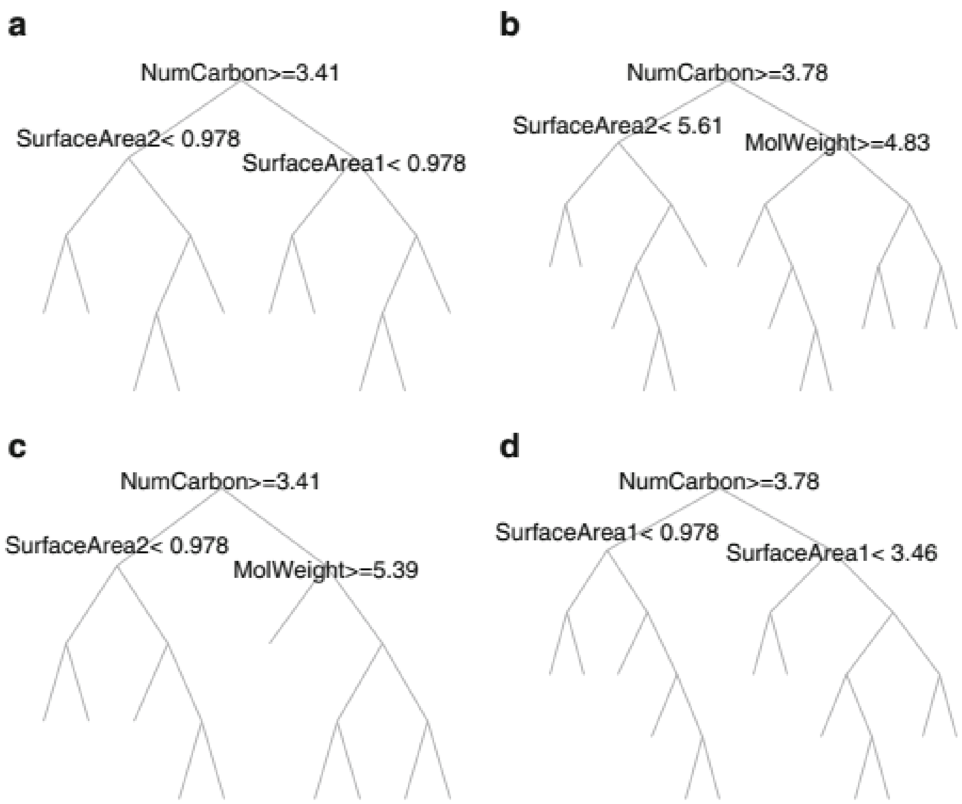
# Topluluk Yöntemleri

## Rasgele Ormanlar

**Amaç:** Ağaçlar arasındaki korelasyonu azaltmak için dalları ayırırken tüm değişkenler yerine sadece rassal sayıda değişkeni kullanmak

Neden?

*

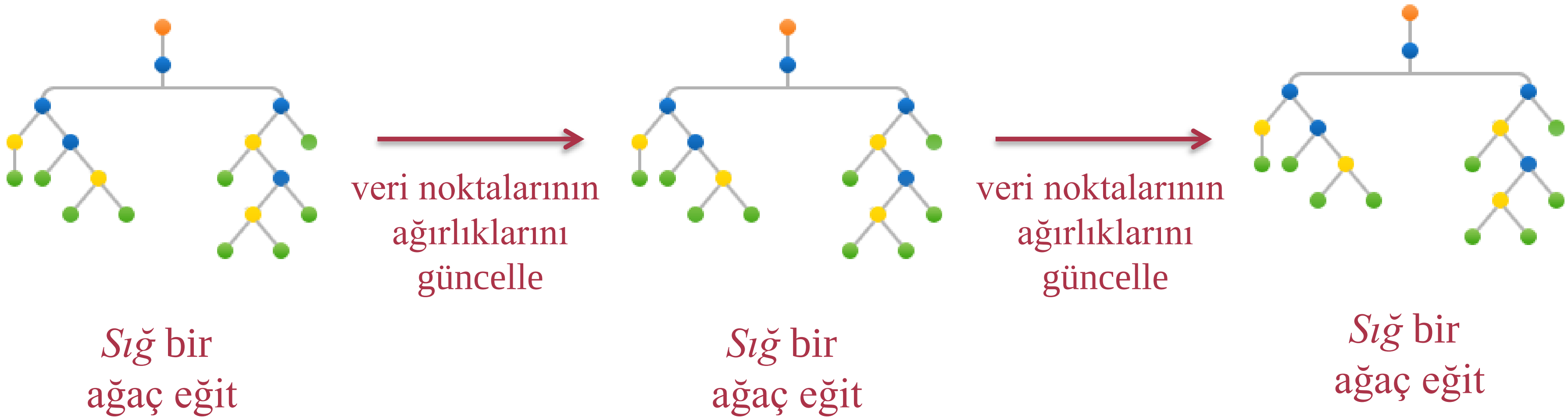



**Applied Predictive Modeling*, M. Kuhn, K. Johnson., Springer, 2013, sf. 195.

# Topluluk Yöntemleri

## Takviye

**Fikir:** Fazla etkin olmayan sınıflandırıcıları, ağırlıklı veri örnekleme tekniğini kullanarak bir araya getirerek daha etkin bir sınıflandırıcı elde etmek




**Güncelleme Kuralı:** Yanlış sınıflandırılan veri noktalarının ağırlıklarını artır

# Ensemble Methods

## **AdaBoost** – İkili Sınıflandırma{+1, -1}

Her bir veri satırı başlangıçta aynı ağırlığa sahip: (*1/n*)

**for** *k=1 to K* **do**

ağırlıklı verileri kullanarak *d* dallı bir ağaç eğit ve and yanlış sınıflandırma hatasını hesapla ($\epsilon_k$)

Ağırlık değerini hesapla $\ln \dfrac{1-\epsilon_k}{\epsilon_k}$

Ağırlıklı verileri güncelle – yanlış sınıflandırılan verilere daha fazla ağırlık ver

**end**

Her bir veri için *k*. aşamadaki değer ile *k*. model tahminini çarparak takviyeli sınıflandırıcının tahminlerini hesapla ve bu miktarları *k*'ye ekle. Eğer toplam pozitif ise veriyi +1 olarak sınıflandır, değilse -1.

Kitaptaki Algoritma 8.2 **bağlanım ağaçları** için takviye örneği veriyor.

# Topluluk Yöntemleri

## **AdaBoost** – İkili Sınıflandırma{+1, -1}

**Algoritma 1:** AdaBoost

1 Başlangıç ağırlıklarını belirle: $w_i^1 = \frac{1}{n}, \quad i = 1, \ldots, n$

2 **for** $k = 1, \ldots, K$ **do**

3 $y_k(x)$ sınıflandırıcısını şu ağırlıklı sınıflandırma hata fonksiyonu ile eğit:

$$\sum_{i=1}^{n} w_i^k I(y_k(x_i) \neq y_i)$$

Yanlış sınıflandırılan veri noktalarının oransal ağırlığını hesapla:

$$\varepsilon^k = \frac{\sum_{i=1}^n w_i^k I(y_k(x_i) \neq y_i)}{\sum_{i=1}^n w_i^k}$$

4 Güncelleme parametresini belirle: $\alpha_k = \ln \frac{1-\varepsilon_k}{\varepsilon_k}$

5 Ağırlıkları güncelle: $w_i^{k+1} = w_i^k e^{\alpha_k I(y_k(x_i) \neq y_i)}, \quad i = 1, \ldots, n$

6 **Çıktı:** $\sum_{k=1}^K \alpha_k y_k(x)$ değerinin işareti (+1 ya da -1)

(AdaBoost Özeti)

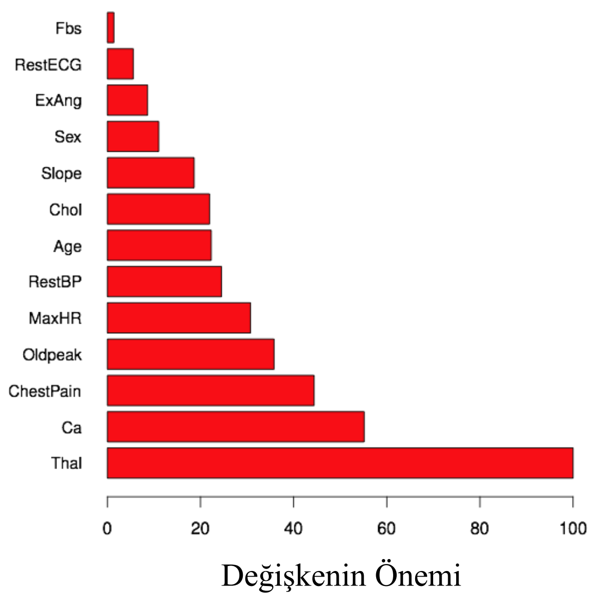
# Topluluk Yöntemleri

- Takviye ve torbalama yöntemleri başka yöntemler ile de uygulanabilirler

- Torbalama yöntemi paralel uygulama için son derece uygun olmasına rağmen takviye yöntemi sıralı yapısı nedeniyle paralelleştirmeye uygun değildir

- Topluluk yöntemleri ile elde edilen modeli yorumlamak güçtür

- Değişken önemini (variable importance) gösteren grafikler kullanılabilir




Bir **değişkenin önemi** Gini indeksinde ya da entropide elde ettiği ortalama azaltmaya göre belirlenir. Daha sonra değişkenler bu önem sırasına göre oranlanarak sıralanır.

# Pratikte

```
# Required packages
from sklearn.tree import DecisionTreeClassifier
from sklearn.datasets import make_classification
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
```

paketler ve fonksiyonlar

```
# Generate and extract features and target data
X, y = make_classification(n_samples=1000,
                           n_features=4,
                           n_informative=2,
                           n_redundant=0,
                           random_state=0,
                           shuffle=False)
```

verinin yaratılması

```
# Create decision tree and train it
DTClassifier = DecisionTreeClassifier(criterion = 'gini',
                                      splitter = 'best',
                                      max_depth = None,
                                      min_samples_leaf = 5)
DTClassifier.fit(X, y)
DTClassifier.score(X, y)
```

karar ağacı uydurumu ve performansının raporlanması

topluluk yöntemleri

```
# Bagging
bagging = BaggingClassifier(DTClassifier,
                            n_estimators = 20,
                            max_samples = 1.0,
                            bootstrap = True,
                            bootstrap_features = False)
bagging.fit(X, y)
bagging.score(X, y)
```

torbalama yöntemi kullanımı ve performansının raporlanması

```
   ...: [DTClassifier.score(X, y),
   ...:  bagging.score(X, y),
   ...:  RF.score(X,y),
   ...:  ABC.score(X,y)]
Out[2]: [0.97, 0.973, 0.975, 0.983]
```

```
# Random Forest
RF = RandomForestClassifier(criterion = 'gini',
                            n_estimators=10,
                            max_depth=None,
                            min_samples_leaf = 5,
                            random_state=0)
RF.fit(X,y)
RF.score(X,y)
```

rastgele ormanlar yöntemi kullanımı ve performansının raporlanması

```
# Adaboost
ABC = AdaBoostClassifier(n_estimators=100, random_state=0)
ABC.fit(X, y)
ABC.score(X,y)
```

AdaBoost yönteminin kullanımı ve performansının raporlanması

# Destek Vektör Makineleri - DVM
# (Support Vector Machines - SVM)

$$(x_i, y_i), \quad i = 1, 2, \ldots, n$$

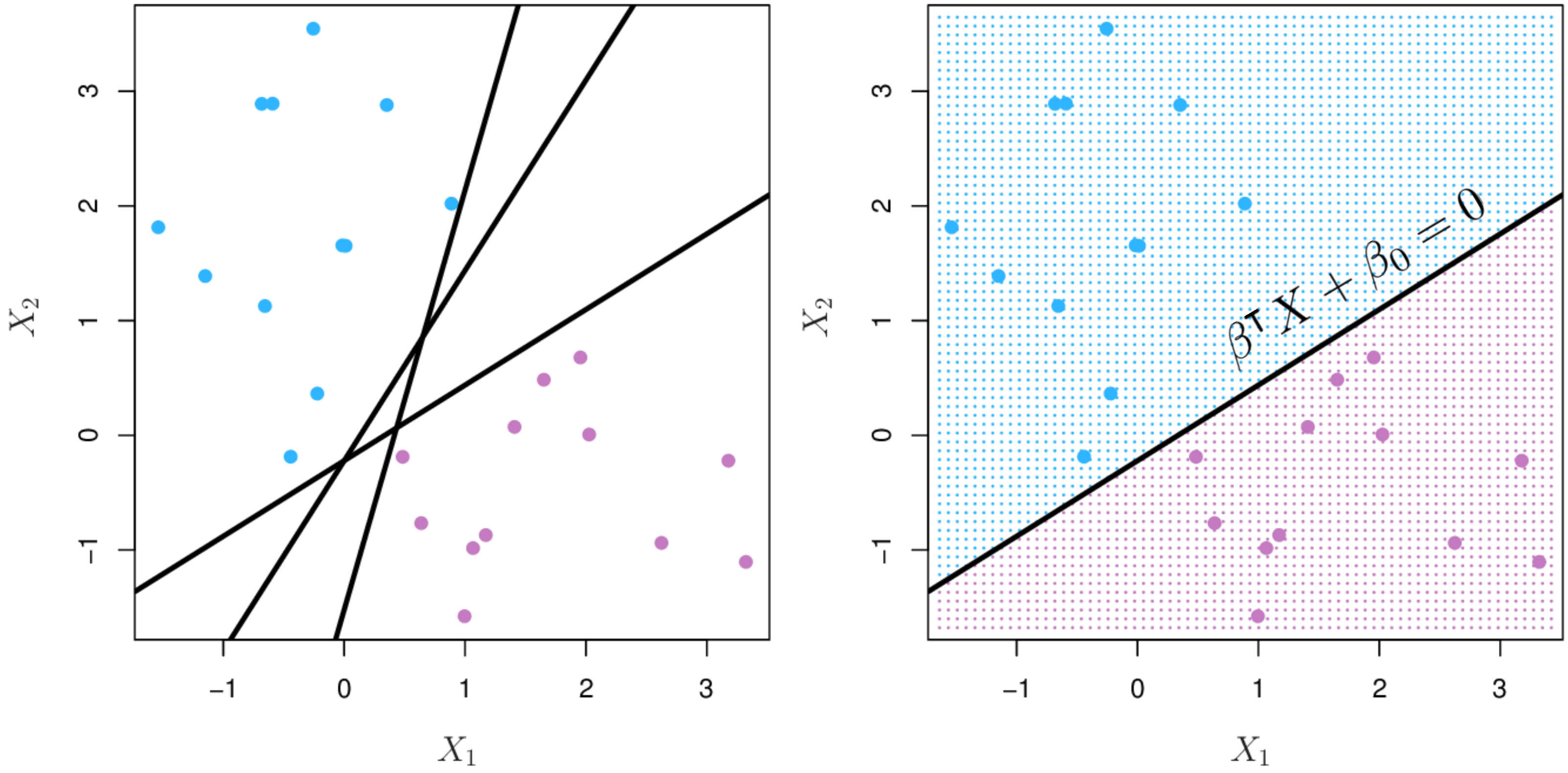
$$x_i \in \mathbb{R}^p \qquad\qquad y_i \in \{-1, +1\}$$

**Fikir:** Bir düzlem (hyperplane) yardımıyla veri noktalarını iki sınıfa ayırmak

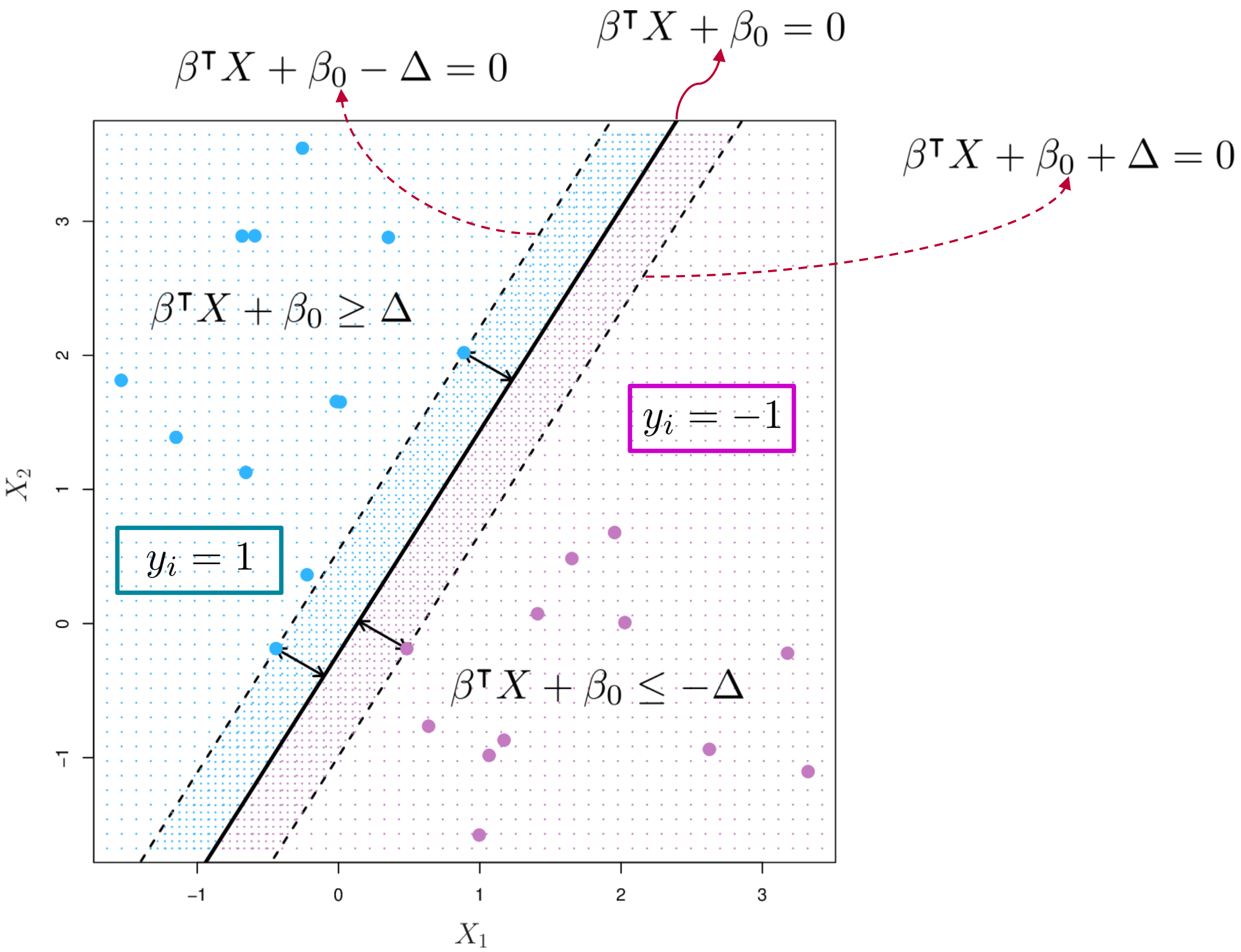
## Düzlem

$$\{X \in \mathbb{R}^p : \beta^\intercal X + \beta_0 = 0,\ \beta \in \mathbb{R}^p, \beta_0 \in \mathbb{R}\}$$

# Tam Ayrıştırılabilir Veri



# Tam Ayrıştırılabilir Veri



# Tam Ayrıştırılabilir Veri

**Amaç:** $\beta$ ve $\Delta$ parametrelerini,

$$\beta^{\intercal} X + \beta_0 - \Delta = 0 \qquad \text{ve} \qquad \beta^{\intercal} X + \beta_0 + \Delta = 0$$

ile gösterilen iki düzlem arasındaki mesafe en fazla olacak şekilde seçmek

Eşitliklerin her iki tarafı $\Delta$ değerine bölünüp, $\Delta = 1$ olarak alınabilir

$$(x_i, y_i) \begin{cases} y_i = 1 \longrightarrow \beta^{\intercal} x_i + \beta_0 - 1 \geq 0 \\ y_i = -1 \longrightarrow \beta^{\intercal} x_i + \beta_0 + 1 \leq 0 \end{cases}$$

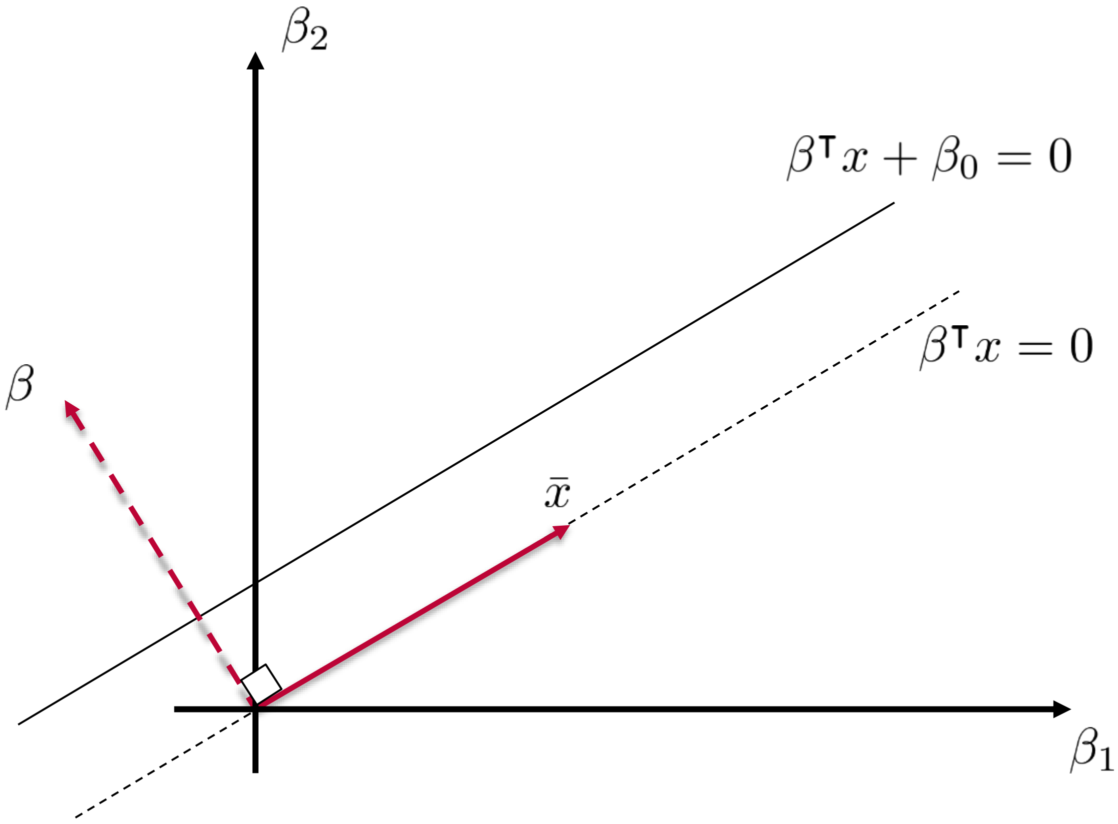
$$y_i(\beta^{\intercal} x_i + \beta_0) \geq 1$$

# Tam Ayrıştırılabilir Veri

**Amaç:** $\beta$ ve $\Delta$ parametrelerini,

$$\beta^{\mathsf{T}} X + \beta_0 - \Delta = 0 \qquad \text{ve} \qquad \beta^{\mathsf{T}} X + \beta_0 + \Delta = 0$$

ile gösterilen iki düzlem arasındaki mesafe en fazla olacak şekilde seçmek




$$\beta = (\beta_1, \beta_2)^{\mathsf{T}}$$

$\beta$ : alt uzaya dik doğrultu

$$\beta^{\mathsf{T}} x = 0$$

Ayrıca $\beta$ vektörü

$$\beta^{\mathsf{T}} x + \beta_0 = 1$$

ve

$$\beta^{\mathsf{T}} x + \beta_0 = -1$$

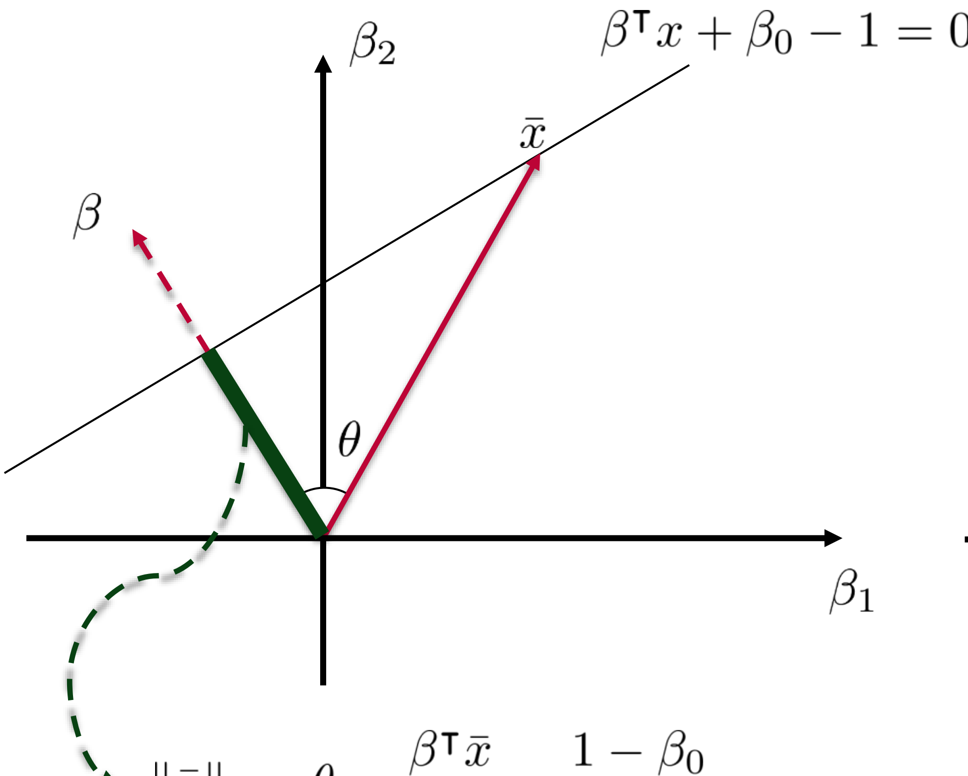
düzlemlerine de dik

# Tam Ayrıştırılabilir Veri

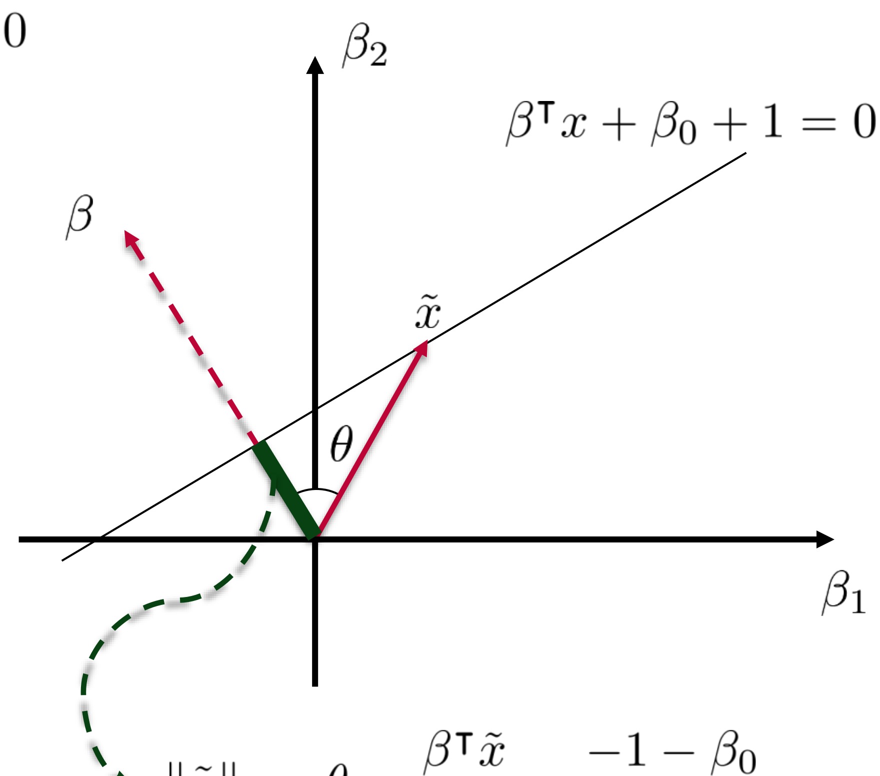
**Amaç:** $\beta$ ve $\Delta$ parametrelerini,

$$\beta^\intercal X + \beta_0 - \Delta = 0 \qquad \text{ve} \qquad \beta^\intercal X + \beta_0 + \Delta = 0$$

ile gösterilen iki düzlem arasındaki mesafe en fazla olacak şekilde seçmek




$$\|\bar{x}\| \cos\theta = \frac{\beta^\intercal \bar{x}}{\|\beta\|} = \frac{1-\beta_0}{\|\beta\|}$$




$$\|\tilde{x}\| \cos\theta = \frac{\beta^\intercal \tilde{x}}{\|\beta\|} = \frac{-1-\beta_0}{\|\beta\|}$$

İki düzlem arasındaki mesafe: $\dfrac{1-\beta_0}{\|\beta\|} - \dfrac{-1-\beta_0}{\|\beta\|} = \dfrac{2}{\|\beta\|}$

# Tam Ayrıştırılabilir Veri

**Amaç:** $\beta$ ve $\Delta$ parametrelerini,

$$\beta^\intercal X + \beta_0 - \Delta = 0 \qquad \text{ve} \qquad \beta^\intercal X + \beta_0 + \Delta = 0$$

ile gösterilen iki düzlem arasındaki mesafe en fazla olacak şekilde seçmek

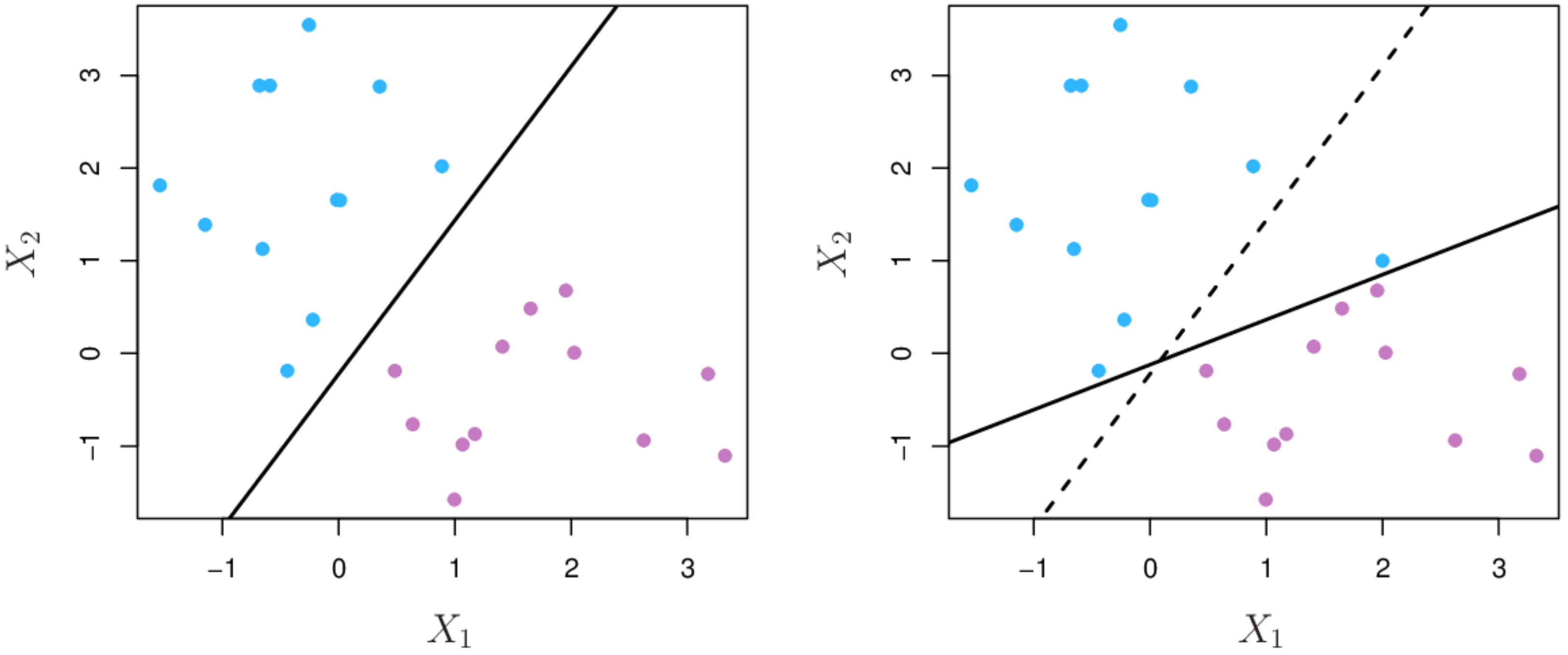
$$\begin{aligned} &\text{enbüyükle} && \frac{2}{\|\beta\|} \\ &\text{öyle ki} && y_i(\beta^\intercal x_i + \beta_0) \geq 1, \quad i = 1, \ldots, n \end{aligned}$$

$$\equiv \begin{aligned} &\text{enküçükle} && \tfrac{1}{2}\|\beta\| \\ &\text{öyle ki} && y_i(\beta^\intercal x_i + \beta_0) \geq 1, \quad i = 1, \ldots, n \end{aligned}$$

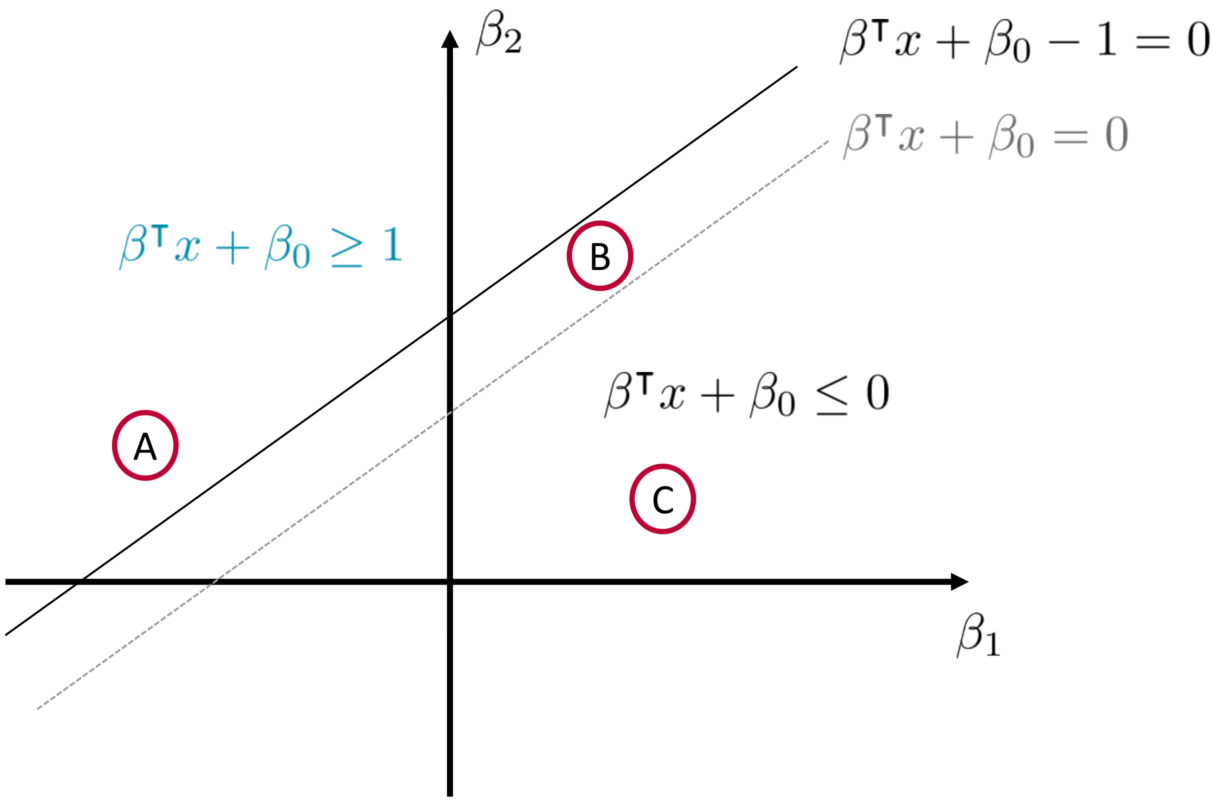
$$\equiv \begin{aligned} &\text{enküçükle} && \tfrac{1}{2}\beta^\intercal\beta \\ &\text{öyle ki} && y_i(\beta^\intercal x_i + \beta_0) \geq 1, \quad i = 1, \ldots, n \end{aligned}$$

Bu matematiksel model, karesel ve dışbükey amaç fonksiyonu ile doğrusal kısıtlardan oluşuyor. Bu yapıdaki modellere *karesel programlama* (quadratic programming) modelleri denir.

# Tam Ayrıştırılabilir Veri



# Tam Ayrıştırılamayan Veri




$(x_i, y_i)$

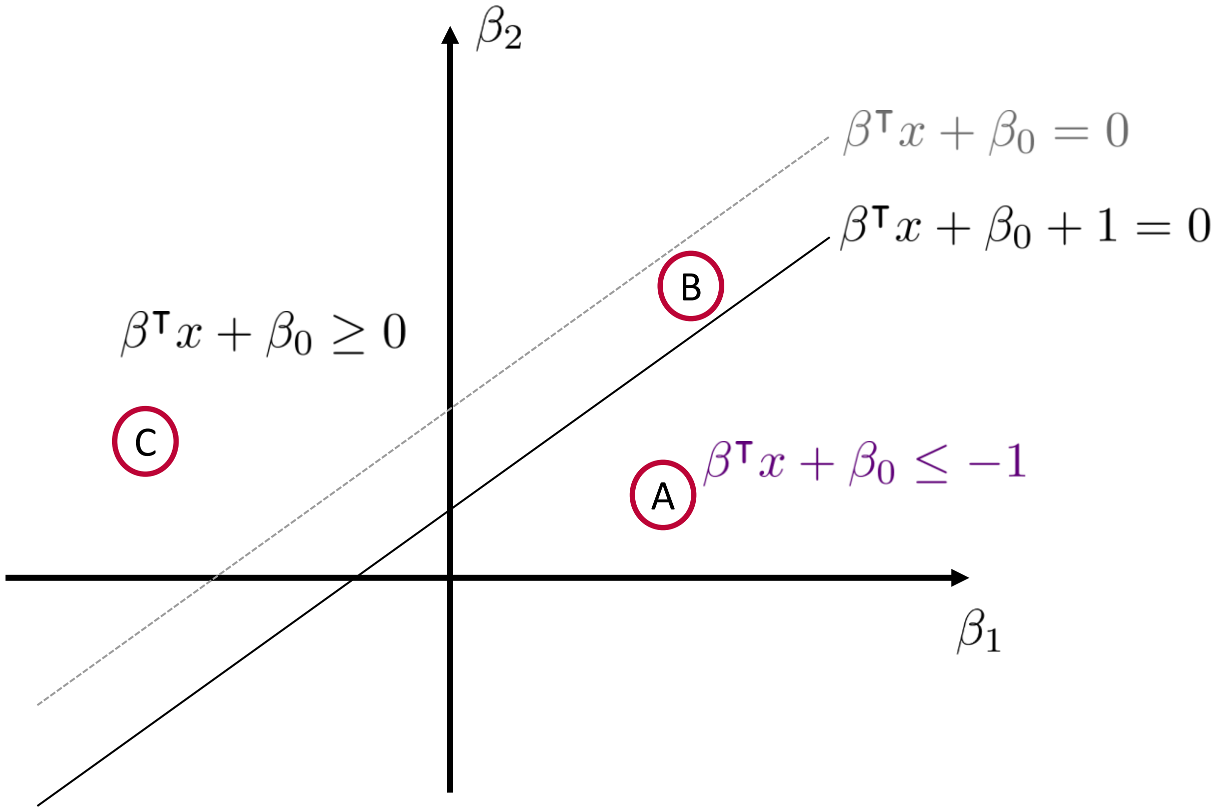
$y_i = 1$

A $\quad y_i(\beta^\mathsf{T}x_i + \beta_0) \geq 1 - \varepsilon, \quad \varepsilon = 0$

B $\quad y_i(\beta^\mathsf{T}x_i + \beta_0) \geq 1 - \varepsilon, \quad 0 < \varepsilon < 1$

C $\quad y_i(\beta^\mathsf{T}x_i + \beta_0) \geq 1 - \varepsilon, \quad \varepsilon \geq 1$

# Tam Ayrıştırılamayan Veri




$(x_i, y_i)$

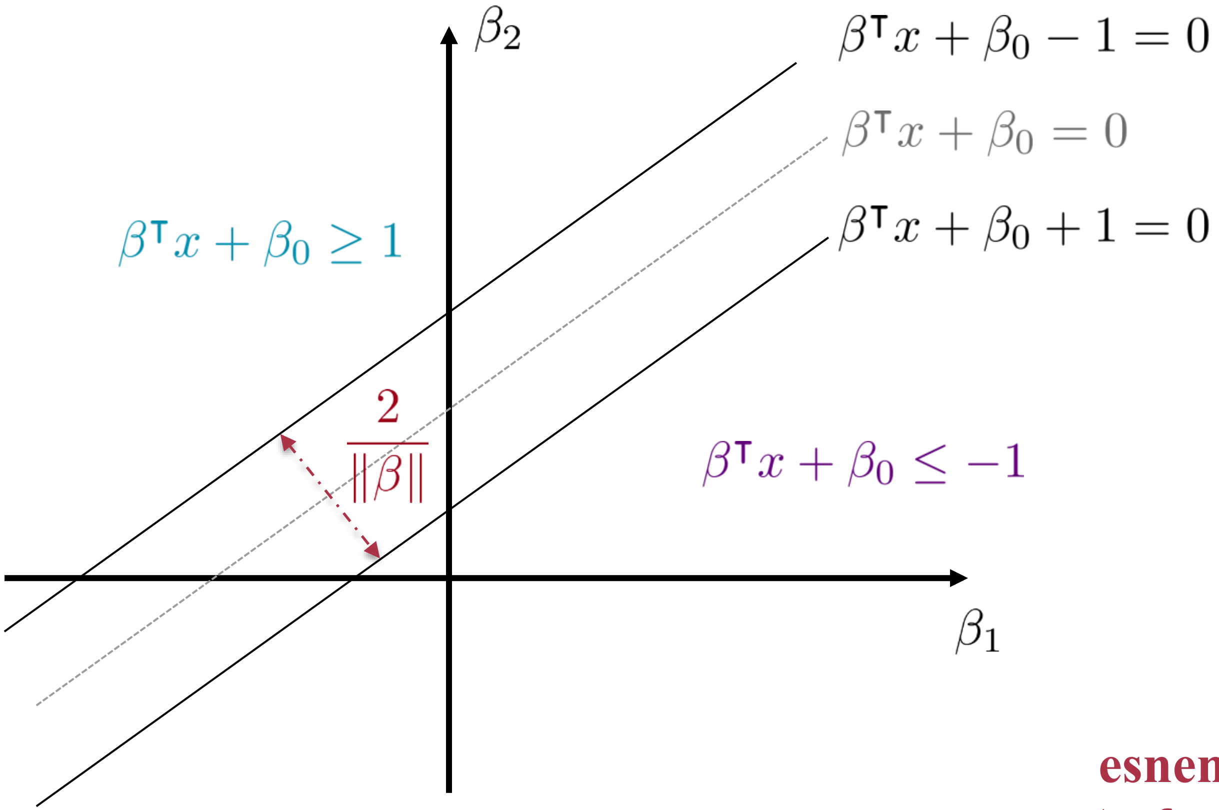
$y_i = -1$

A $\quad y_i(\beta^\top x_i + \beta_0) \geq 1 - \varepsilon, \quad \varepsilon = 0$

B $\quad y_i(\beta^\top x_i + \beta_0) \geq 1 - \varepsilon, \quad 0 < \varepsilon < 1$

C $\quad y_i(\beta^\top x_i + \beta_0) \geq 1 - \varepsilon, \quad \varepsilon \geq 1$

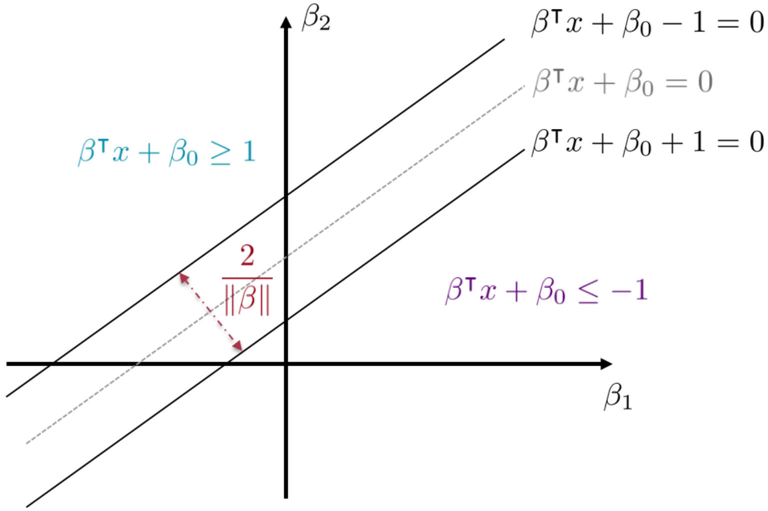
# Tam Ayrıştırılamayan Veri




$$y_i(\beta^\intercal x_i + \beta_0) \geq 1 - \varepsilon, \quad \varepsilon = 0$$
$$y_i(\beta^\intercal x_i + \beta_0) \geq 1 - \varepsilon, \quad 0 < \varepsilon < 1$$
$$y_i(\beta^\intercal x_i + \beta_0) \geq 1 - \varepsilon, \quad \varepsilon \geq 1$$

**esneme payı (soft margin)**

# Tam Ayrıştırılamayan Veri




$y_i(\beta^\intercal x_i + \beta_0) \geq 1 - \varepsilon, \quad \varepsilon = 0$
$y_i(\beta^\intercal x_i + \beta_0) \geq 1 - \varepsilon, \quad 0 < \varepsilon < 1$
$y_i(\beta^\intercal x_i + \beta_0) \geq 1 - \varepsilon, \quad \varepsilon \geq 1$

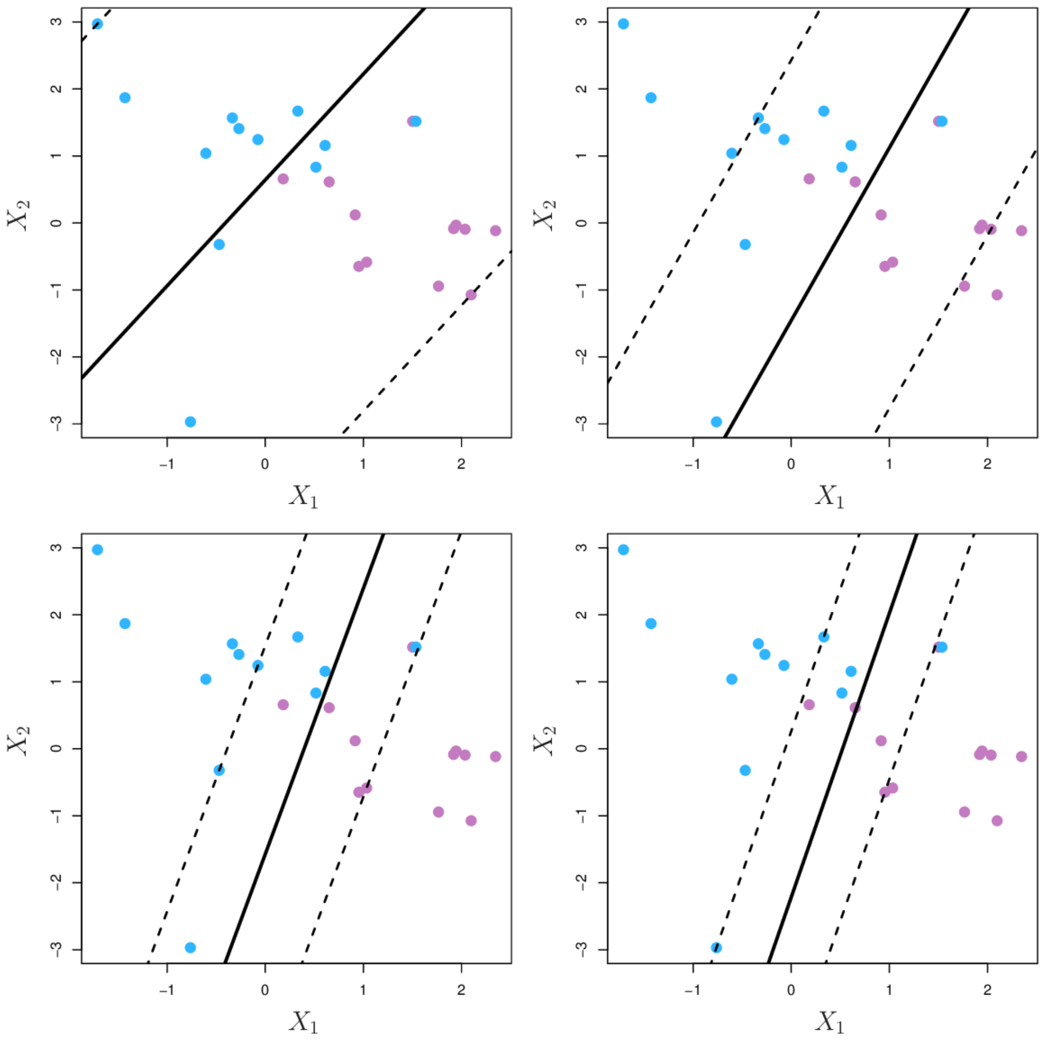
enküçükle $\quad \frac{1}{2}\beta^\intercal\beta + c\sum_{i=1}^{n} \varepsilon_i^d$

öyle ki $\quad y_i(\beta^\intercal x_i + \beta_0) \geq 1 - \varepsilon_i, \quad i = 1, \ldots, n$

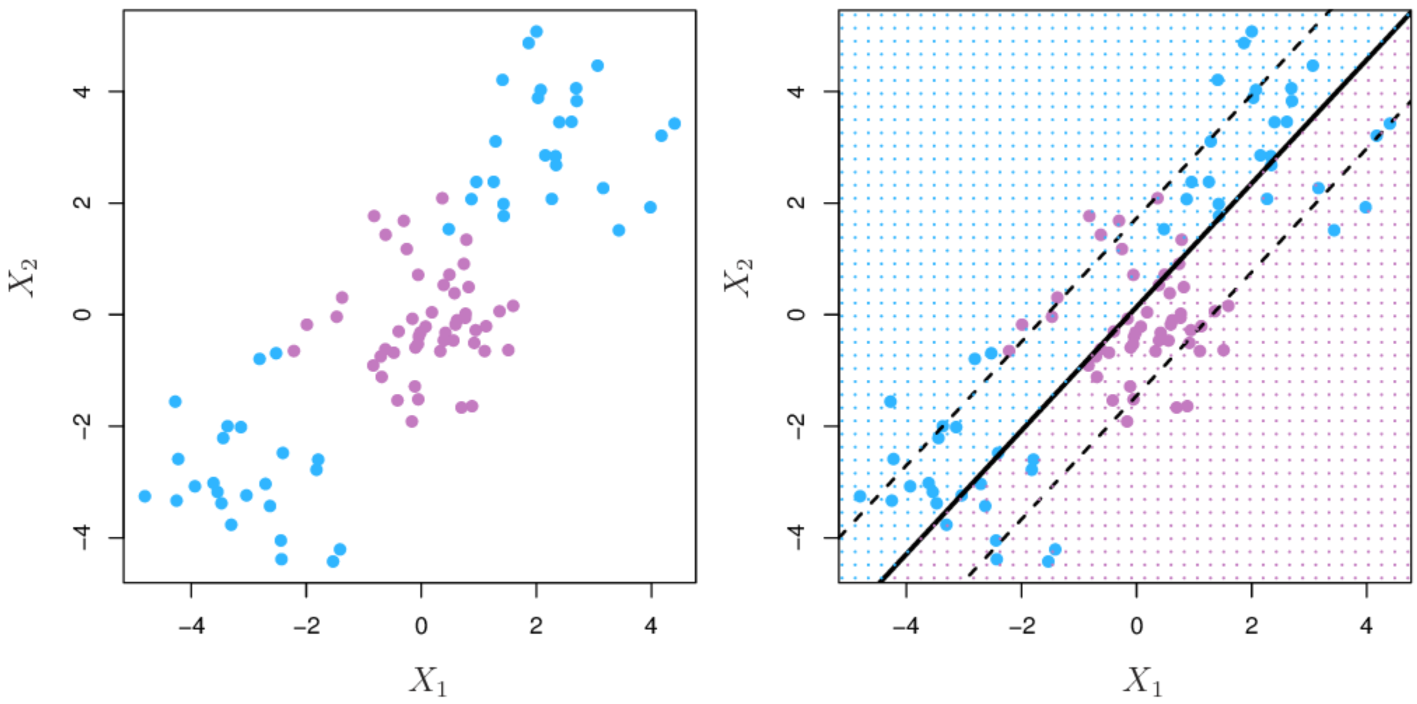
$\varepsilon_i \geq 0, \quad i = 1, \ldots, n$

Burada *c* parametresi çapraz geçerlilik sınaması ile belirlenebilir (uygulamalarda *d* = 1 ve *d* = 2 değerleri sık kullanılır)

# Tam Ayrıştırılamayan Veri



# Çekirdekler (Kernels)



# Çekirdekler

| | |
|---|---|
| enküçükle | $\frac{1}{2}\beta^\intercal\beta$ |
| öyle ki | $y_i(\beta^\intercal x_i + \beta_0) \geq 1, \quad i = 1, \ldots, n$ |

Lagrange çarpanları

Lagrange Fonksiyonu

$$\mathcal{L}(\beta, \beta_0; \alpha) = \frac{1}{2}\beta^\intercal\beta - \sum_{i \in \mathcal{A}} \alpha_i \left(y_i(\beta^\intercal x_i + \beta_0) - 1\right)$$

aktif kısıtlar kümesi

Optimallik Şartları

$$\nabla_\beta \mathcal{L}(\beta, \beta_0; \alpha) = \beta - \sum_{i \in \mathcal{A}} \alpha_i y_i x_i = 0 \implies \beta = \sum_{i \in \mathcal{A}} \alpha_i y_i x_i$$

$$\frac{\partial \mathcal{L}(\beta, \beta_0; \alpha)}{\partial \beta_0} = -\sum_{i \in \mathcal{A}} \alpha_i y_i = 0 \implies \sum_{i \in \mathcal{A}} \alpha_i y_i = 0$$

$$\begin{aligned}
\mathcal{L}(\beta, \beta_0; \alpha) = \quad & \tfrac{1}{2}\left(\textstyle\sum_{i \in \mathcal{A}} \alpha_i y_i x_i\right)^\intercal \left(\textstyle\sum_{j \in \mathcal{A}} \alpha_j y_j x_j\right) - \left(\textstyle\sum_{i \in \mathcal{A}} \alpha_i y_i \left(\textstyle\sum_{j \in \mathcal{A}} \alpha_j y_j x_j\right)^\intercal x_i\right) \\
& - \textstyle\sum_{i \in \mathcal{A}} \alpha_i y_i \beta_0 + \textstyle\sum_{i \in \mathcal{A}} \alpha_i \\
= \quad & \textstyle\sum_{i \in \mathcal{A}} \alpha_i - \tfrac{1}{2} \textstyle\sum_{i \in \mathcal{A}} \textstyle\sum_{j \in \mathcal{A}} \alpha_i \alpha_j y_i y_j \boxed{x_i^\intercal x_j}
\end{aligned}$$

!

# Çekirdekler

$$\mathcal{L}(\beta, \beta_0; \alpha) = \sum_{i \in \mathcal{A}} \alpha_i - \frac{1}{2} \sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{A}} \alpha_i \alpha_j y_i y_j \boxed{x_i^\intercal x_j}^{*}$$

**tek yapmamız gereken bu iç çarpımları hesaplamak**

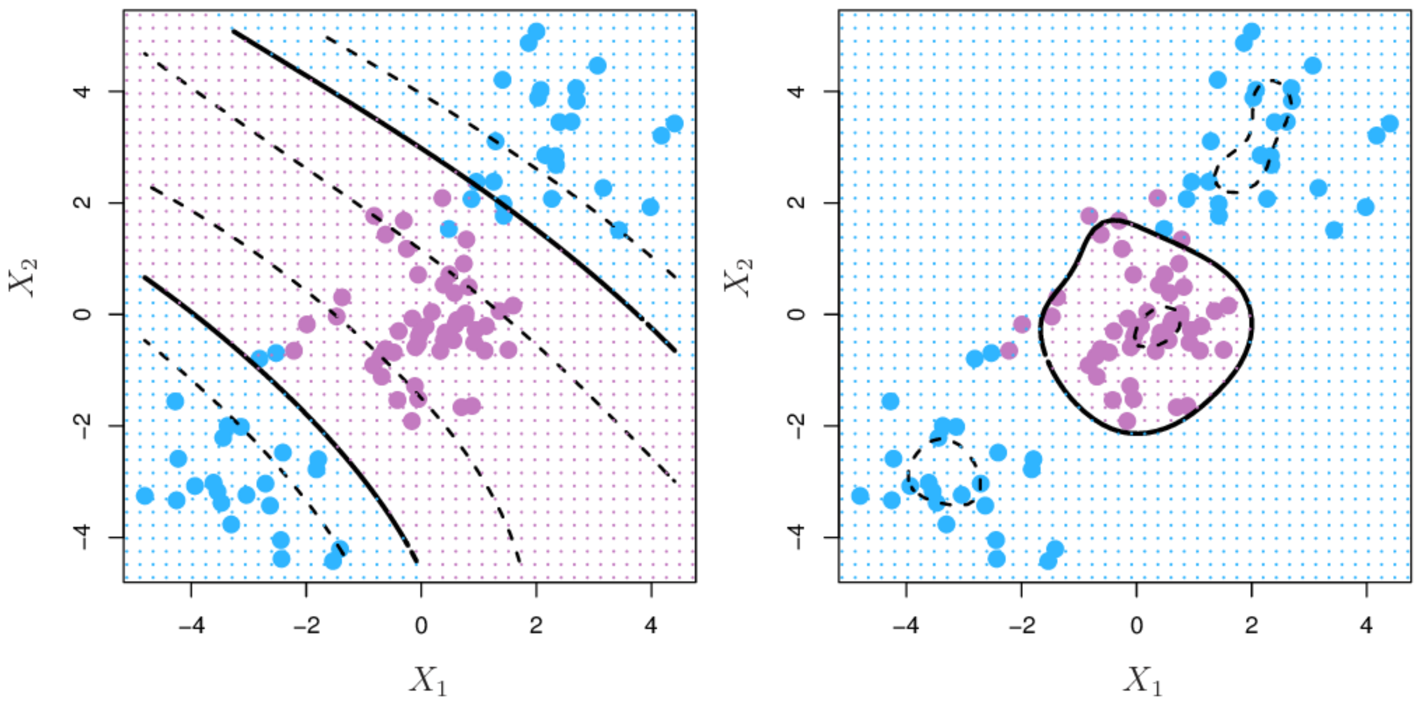
**Fikir:** İç çarpım hesabını daha yüksek boyutlu uzayda iki veri arasındaki benzerliği ölçen bir çekirdek ile değiştirmek

Doğrusal Çekirdek* $\quad K(x_i, x_j) = x_i^\intercal x_j$

Polinom Çekirdek $\quad K(x_i, x_j) = (1 + x_i^\intercal x_j)^d$

Radyal Çekirdek $\quad K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}, \;\; \gamma > 0$

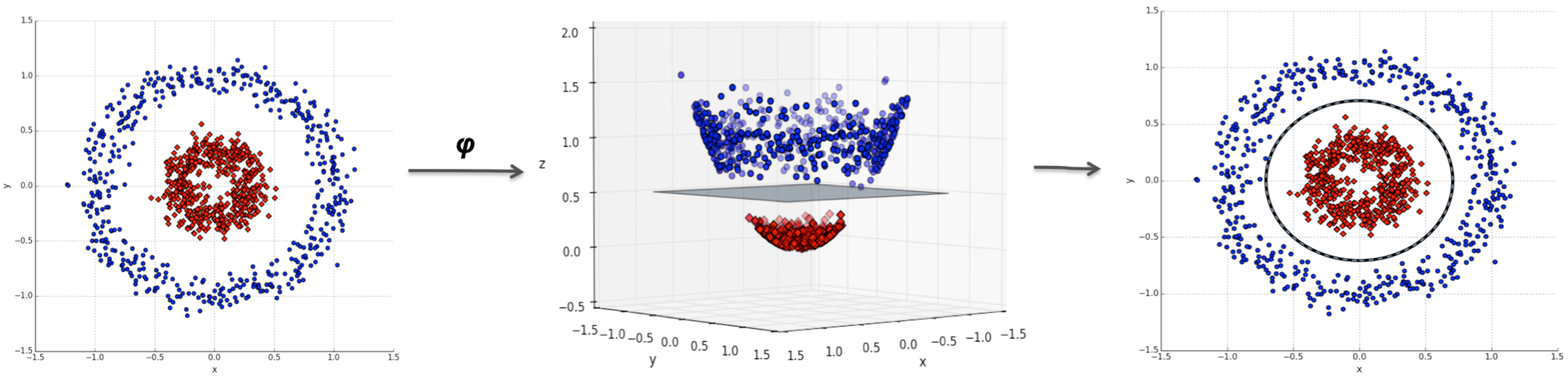
# Çekirdekler



# Çekirdekler

$$\varphi : \mathbb{R}^p \to \mathbb{R}^q$$

$$K(x_i, x_j) = \varphi(x_i)^\intercal \varphi(x_j)$$



# Çekirdekler

Polinom çekirdek: $K(x_i, x_j) = (1 + x_i^\intercal x_j)^d$

$$x_i, x_j \in \mathbb{R}^2$$
$$d = 2$$

$$K(x_i, x_j) = (1 + x_i^\intercal x_j)^2 = \quad 1 + x_{i1}^2 x_{j1}^2 + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2} + 2x_{i1}x_{i2}x_{j1}x_{j2}$$

$$\varphi : \mathbb{R}^2 \to \mathbb{R}^6$$

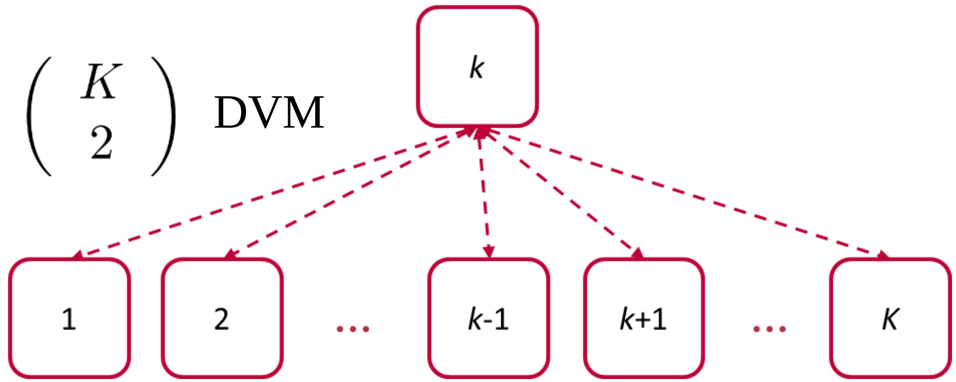
$$\varphi(x_i) = (1, x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}, \sqrt{2}x_{i1}x_{i2})^\intercal$$

$$K(x_i, x_j) = \varphi(x_i)^\intercal \varphi(x_j) = (1 + x_i^\intercal x_j)^2$$

# Çoklu Sınıf

Tüm Çiftler

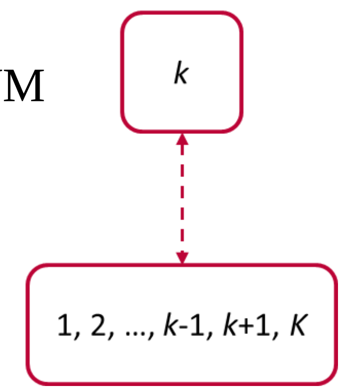
$\binom{K}{2}$ DVM




$x_0$ en sık atandığı sınıfta kabul edilir

Biri Diğerlerine Karşı

$K$ DVM




$$\beta_0^{(k)} + \beta^{(k)\intercal} x_0$$

$$\left.\begin{array}{c} \beta_0^{(1)} + \beta^{(1)\intercal} x_0 \\ \beta_0^{(2)} + \beta^{(2)\intercal} x_0 \\ \vdots \\ \beta_0^{(K)} + \beta^{(K)\intercal} x_0 \end{array}\right\}$$

$x_0$ en yüksek değeri alan sınıfa atanır

# Özet

- Lojistik Bağlanım
- Doğrusal Ayrımlayıcı Çözümleme
- Sonuçların değerlendirilmesi
- Karar Ağaçları
- Bağlanım ve Sınıflandırma Ağaçları
- Topluluk yöntemleri
- Destek Vektör Makineleri