



# VERİ ANALİZİ OKU

2025-2026

Makine Öğrenmesi - 2

Dr. Öğr. Üyesi M. Fuat KINA

Marmara Üniversitesi

Nüfus ve Sosyal Araştırmalar Enstitüsü



## 1. İleri Makine Öğrenmesi Modelleri

- Naive Bayes Sınıflandırma
- Support Vector Machines
- Decision Trees (Karar Ağaçları) and Random Forests
- Principal Component Analysis (Temel Bileşen Analizi)

## 2. Derin Öğrenme

- Neural networks (Sinir Ağları)

# 1. İleri Makine Öğrenmesi

# Naif Bayes Sınıflandırma

- **Gözetimli öğrenme:** Sınıflandırıcı (Classifier)
- **Bayes** teoremine dayanır (naif varsayımlı ile).
- Gaussian, Multinomial, Complement, Bernoulli...

$$P(\text{parameters}|\text{data}) = \frac{P(\text{data}|\text{parameters}) * P(\text{parameters})}{P(\text{data})}$$

$P(\text{parameters}|\text{data})$  -> posterior distribution what we know about parameters after we see the data

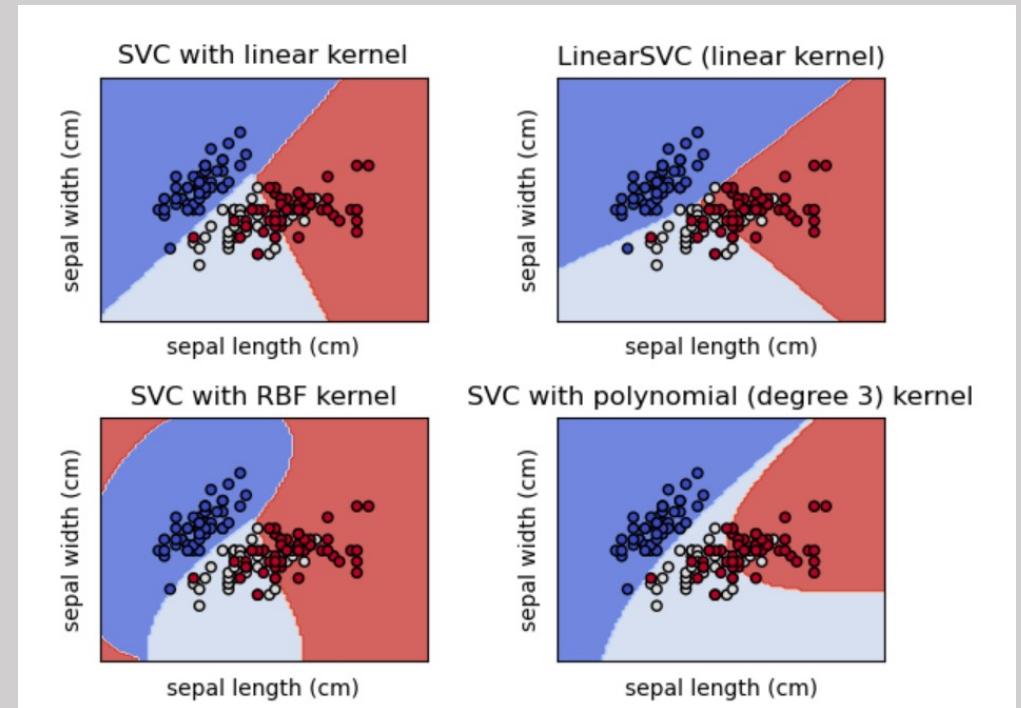
$P(\text{parameters})$  -> prior distribution what we know about parameters

$P(\text{data}|\text{parameters})$  -> likelihood the data according to statistical model



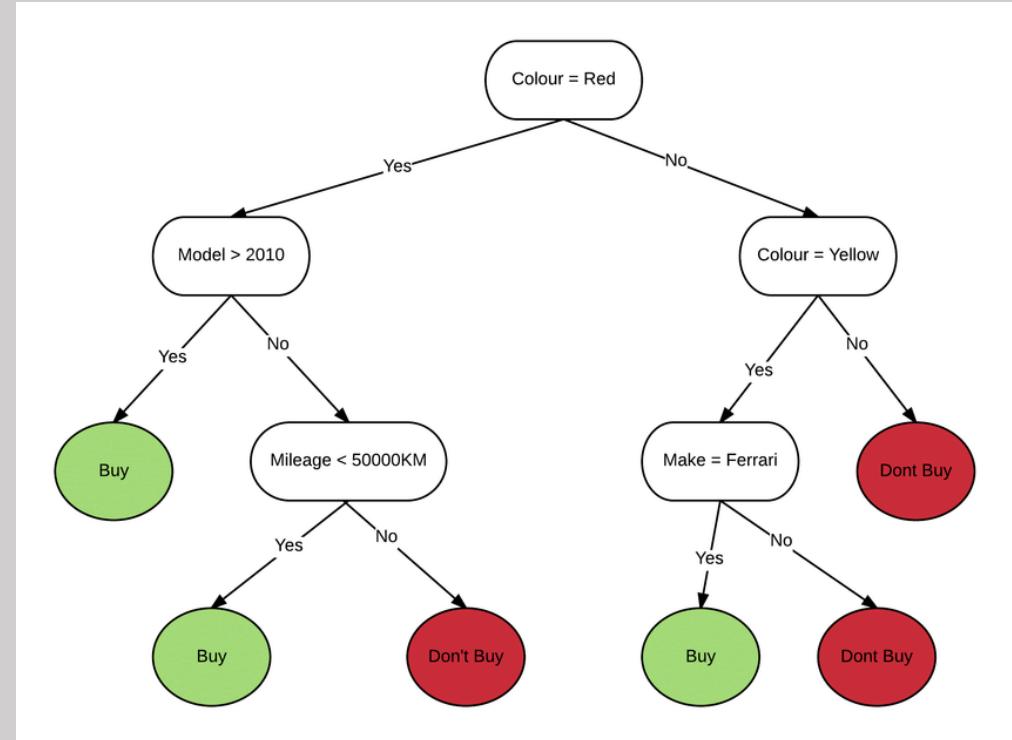
# Support Vector Machines

- **Gözetimli Öğrenme: Sınıflandırıcı (Classifier)**  
**SVM (Support Vector Machine)**, eğitim örneklerini uzaydaki noktalara eşler ve iki kategori arasındaki **boşluğun (margin)** genişliğini **maksimize** etmeye çalışır.
- **N-boyutlu bir uzay** düşün.
- İki gruptaki **en yakın noktalara** olan mesafeyi mümkün olduğunca **büyük** tutacak şekilde bir sınır (ayırıcı hiper-düzlem) bulur.
- **Kernel** seçebilirsin: **linear, polynomial, RBF...**



# Decision Trees (Kar Ağacıları) and Random Forests

- **Gözetimli öğrenme.** Amaç, veri özelliklerinden çıkarılan **basit karar kurallarını** öğrenerek bir **hedef değişkenin** değerini tahmin eden bir model oluşturmak. Bir ağaç, **parça parça sabit (piecewise constant)** bir yaklaşım/approximation olarak düşünülebilir.
- Hem **kategorik** hem de **sürekli** hedef değişkenlerle çalışır.
  - Random forest için: ağaç sayısını (number of trees) belirle.
  - Bölme (split) kriterleri: Gini, entropy, chi-square
  - Overfitting'i önlemek için ağaçın maksimum derinliğini (dikey derinlik) belirle. Derinlik arttıkça model, belirli bir örneklemeye çok özgü ilişkileri öğrenebilir; bu yüzden CV (cross-validation) ile ayarlanmalıdır.



# Principal Component Analysis (Temel Bileşen Analizi)

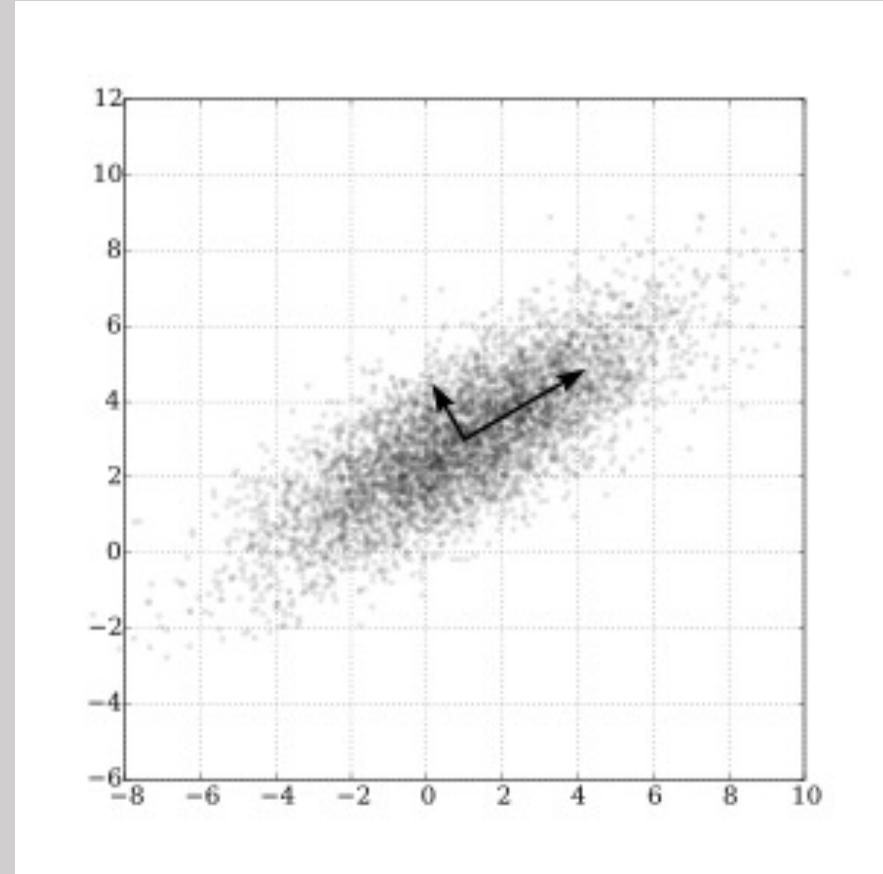
- **Gözetimsiz öğrenme**

Verinin **temel bileşenlerini (principal components)** hesaplama ve bu bileşenleri kullanarak veride **baz (basis) dönüşümü** yapma sürecidir.

- **Varyansı maksimize eder.**

- **Orthogonal** boyutlar çıkarır.

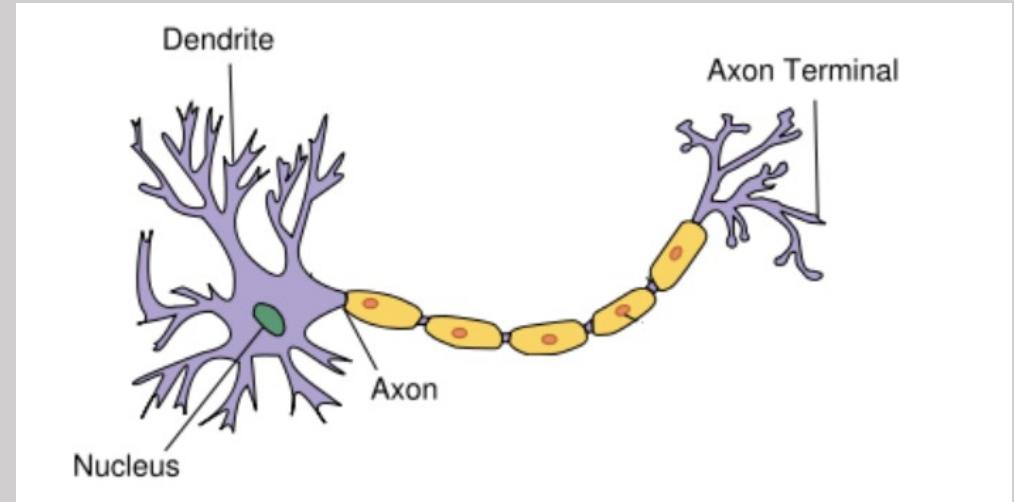
- Verinin boyutlarını **dekorele eder**: PCA ile elde edilen özellikler **birbiriyle korelasyonlu değildir**.



# 2. Derin Öğrenmeye Giriş

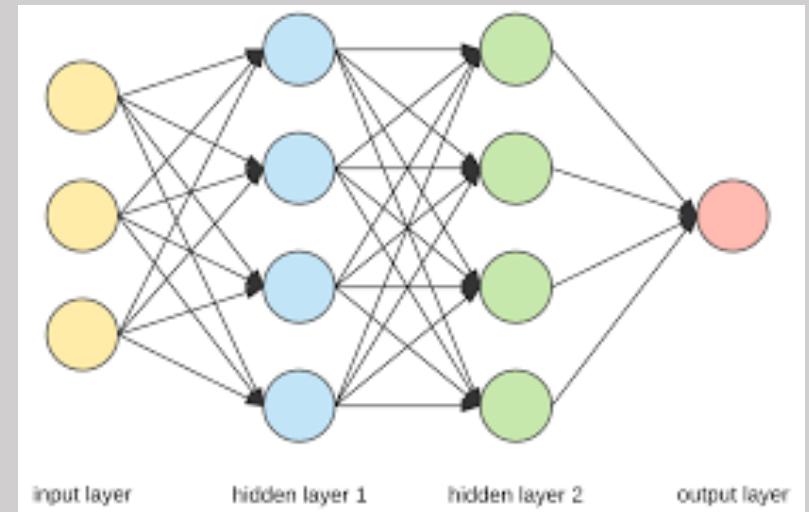
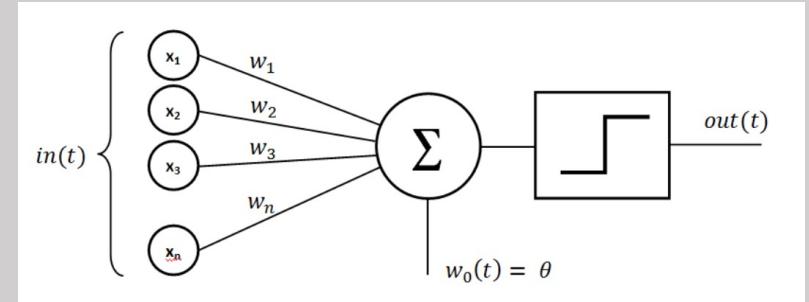
# Neural network (Sinir Ağları)

- **Temel amaç:** İnsan beynine benzer şekilde örüntüleri/nesneleri tanımak.
- **İnsan sinir sisteminden** esinlenmiştir.
- **Gözetimli öğrenme** kapsamında kullanılabilir.
  - Python'da **scikit-learn** ile kolayca çalıştırılabilir.
  - Özellikle farklı türde özellikleri (feature) birlikte analiz etmek istediğiinde **çok verimlidir**.



# Neural network (Sinir Ağları)

- En basit sinir ağı, yalnızca **tek bir nöron**dan oluşur ve yukarıdaki şekilde gösterildiği gibi **perceptron** olarak adlandırılır.
- Soldaki **ilk düğüm (node) grubu**, verideki **ozellik (feature) sayısı** ile aynıdır.
- Yapay Sinir Ağı (Çok Katmanlı Algılayıcı / Multilayer Perceptron, MLP)**



- **İleri yayılım (Forward Propagation):** Girdiyi ağırlıklandıırıp aktivasyon fonksiyonunu çalıştırarak **çıktı katmanına** ulaşma süreci
  - Girdi katmanına gelen değerler **ağırlıklarla çarpılır**, üzerine bir **bias (sabit terim)** eklenir.
  - İlk gizli katmanda nöronlar, girdi katmanından gelen değer üzerinde çalışan bir **aktivasyon fonksiyonuna** sahiptir.
  - İlk gizli katmandan çıkan çıktılar, ikinci gizli katmanın **ağırlıklarıyla** çarpılır (katman katman ilerler).
- **Geri yayılım (Back Propagation)**
  - Hata, **tahmin edilen çıktı ile istenen (gerçek) çıktı** arasındaki fark ölçülerek hesaplanır.
  - Amaç: **hatayı minimize etmek!!!**

# İpç Sinir Ağları (Uygulama - Sklearn and Keras)

---

- **Mimarîyi (architecture) belirle**
  - Kaç katman istiyorsun? Her katmanda kaç düğüm (node) olacak?
  - Her katmanda hangi **aktivasyon fonksiyonunu** kullanmak istiyorsun?
    - Identity (kimlik) fonksiyonu, ReLU, softmax vb.
- **Compile (derleme)**
  - Bu adım **kayıp (loss) fonksiyonunu** ve optimizasyonun nasıl çalışacağına dair bazı detayları belirler.
  - Kayıp fonksiyonu: **Mean Squared Error (MSE), categorical\_crossentropy**
  - Optimizasyon algoritması (optimizer): **Adam**
- **Modeli eğit (fit the model)**
- **Sonucu tahmin et (predict the outcome)**

