

# Denetimli Öğrenme (Supervised Learning)

Denetimli öğrenme, makine öğrenmesinin temel yaklaşımlarından biridir ve önceden etiketlenmiş veri setleri üzerinde çalışır. Bu teknik, girdi ve çıktı ilişkilerini öğrenerek, veri setindeki desenleri tespit etmeyi amaçlar. Girdi verileri (bağımsız değişkenler) ile bu verilere karşılık gelen doğru çıktılar (bağımlı değişkenler veya etiketler) arasındaki ilişkiyi matematiksel olarak modellemeye çalışır.

## Denetimli Öğrenme Süreci

- Veri Toplama ve Hazırlama:** İlk adım, problem alanına uygun verilerin toplanması ve bu verilerin temizlenmesi, dönüştürülmesi veya özellik mühendisliği teknikleri ile hazırlanmasıdır. Bu adım, veri setinin kalitesi ve uygunluğu açısından kritik öneme sahiptir.
- Veri Bölme:** Toplanan veri seti genellikle eğitim ve test seti olarak ikiye bölünür. Eğitim seti, modelin öğrenmesi için kullanılırken, test seti ise modelin performansının değerlendirilmesinde kullanılır. Ayrıca, çapraz doğrulama gibi teknikler de kullanılabilir.
- Model Seçimi:** Veri setinin özelliklerine ve problem türüne göre uygun bir makine öğrenmesi algoritması seçilir. Örneğin, lineer regresyon sürekli bir çıktı tahmini yaparken, karar ağaçları sınıflandırma veya regresyon problemleri için kullanılabilir.
- Model Eğitimi:** Seçilen algoritma, eğitim veri seti üzerinde çalıştırılır ve girdi-veri çiftleri arasındaki ilişkiyi matematiksel olarak modellemeye başlar. Bu süreçte, algoritma parametreleri ve ağırlıkları, eğitim veri setine göre ayarlanır.
- Model Değerlendirme:** Eğitilen model, test veri seti üzerinde değerlendirilir. Performans metrikleri (doğruluk, hassasiyet, geri çağırma, F1 skoru vb.) kullanılarak modelin başarısı ölçülür. Bu adım, modelin genelleme yeteneğini değerlendirmek için kritik öneme sahiptir.
- Model İyileştirme:** Modelin performansını artırmak için hiperparametre optimizasyonu, özellik seçimi, ensemble yöntemleri (örneğin, rastgele ormanlar veya gradyan artışı makineleri) ve veri artırma teknikleri gibi yöntemler kullanılabilir.
- Model Kullanımı:** Başarılı bir şekilde eğitilen model, gerçek dünya verileri üzerinde kullanılmak üzere hazır hale getirilir. Uygulama alanına bağlı olarak, modelin sürekli olarak güncellenmesi veya yeniden eğitilmesi gerekebilir.

## Denetimli Öğrenme Algoritmaları ve Uygulama Alanları

- Lineer Regresyon:** Sürekli bir çıktı değişkenini tahmin etmek için kullanılır. Örneğin, ev fiyatları gibi sürekli değişkenlerin tahmini için idealdir.
- Lojistik Regresyon:** İkili sınıflandırma problemlerinde kullanılır. Örneğin, bir hasta için bir hastalığın var olup olmadığını tahmin etmek için kullanılabilir.
- Karar Ağaçları ve Rastgele Ormanlar:** Sınıflandırma veya regresyon problemlerinde geniş bir kullanım alanına sahiptir. Özellikle özellik seçimi ve veri anlayışında güçlüdürler.
- Destek Vektör Makineleri (SVM):** Hem sınıflandırma hem de regresyon problemlerinde kullanılır. Özellikle doğrusal olmayan veri setlerinde etkili sonuçlar verebilir.
- Yapay Sinir Ağları:** Derin öğrenme alanında popülerdir ve görüntü ve ses tanıma gibi karmaşık görevlerde kullanılır.

# Denetimli Öğrenmenin Avantajları ve Dezavantajları

## Avantajlar

- **Doğru Sonuçlar:** Etiketlenmiş veri setleri üzerinde eğitildiği için doğru sonuçlar elde eder.
- **Geniş Uygulama Alanı:** Farklı türde problemler için uygundur (sınıflandırma, regresyon, sıralama vb.).
- **Anlaşılabilirlik:** Birçok algoritma, sonuçların nasıl üretildiğini açıkça gösterir.

## Dezavantajlar

- **Veri Gereksinimi:** Eğitim için büyük miktarda etiketlenmiş veri gerektirir.
- **Overfitting Riski:** Modelin eğitim veri setine fazla uyum sağlaması ve yeni verilerde performansının düşmesi riski vardır.
- **Hiperparametre Ayarı:** Optimal performans için hiperparametrelerin doğru şekilde ayarlanması gereklidir.

## Sonuç

Denetimli öğrenme, bilgisayarların veri analizinde ve karar verme süreçlerinde önemli bir rol oynayan güçlü bir tekniktir. Doğru veri setleri, uygun algoritma seçimi ve dikkatli model değerlendirmesi ile başarılı sonuçlar elde edilebilir. Her problem için en uygun yaklaşımın belirlenmesi ve sürecin dikkatlice yönetilmesi, denetimli öğrenme uygulamalarının etkinliğini artırır.



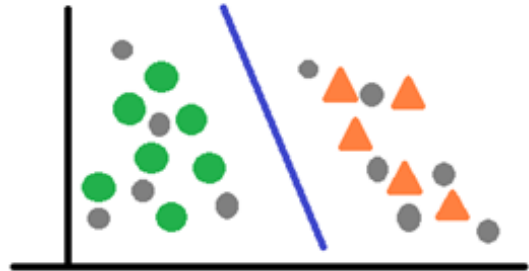
a) Etiketli Veri



b) Etiketli ve Etiketsiz Veri



c) Denetimli Öğrenme



d) Yarı Denetimli Öğrenme

## Örnek

Bu örnekte, popüler bir veri seti olan Iris veri setini kullanacağız. Iris veri seti, üç farklı türde iris bitkisinin çiçek yapraklarıyla ilgili dört özellik içerir ve bu özelliklere dayanarak bitki türlerini sınıflandırmayı amaçlar.

Bu örnek, scikit-learn kütüphanesini kullanarak bir karar ağacı (Decision Tree) sınıflandırıcısı oluşturacaktır.

```
# Gerekli kütüphanelerin import edilmesi
import numpy as np
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

# Iris veri setinin yüklenmesi
iris = load_iris()
X = iris.data # Özellikler (features)
y = iris.target # Etiketler (labels)

# Eğitim ve test setlerine ayırma (70% eğitim, 30% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Karar ağacı sınıflandırıcısının oluşturulması
clf = DecisionTreeClassifier()

# Modelin eğitim verisi ile eğitilmesi
clf.fit(X_train, y_train)

# Test verisi ile tahmin yapma
y_pred = clf.predict(X_test)
```

```
# Modelin doğruluk (accuracy) skorunun hesaplanması
accuracy = accuracy_score(y_test, y_pred)

# Sonuçların yazdırılması
print(f"Model Doğruluğu: {accuracy * 100:.2f}%")
```

## Açıklamalar

### 1. Veri Setinin Yüklenmesi:

- o `load_iris()` fonksiyonu ile Iris veri seti yüklenir.

- x değişkeni, çiçek yapraklarının uzunluk ve genişlik bilgilerini içerir (features).
  - y değişkeni ise çiçek türlerini (labels) içerir.
2. **Veri Setinin Eğitim ve Test Setlerine Ayrılması:**
    - `train_test_split()` fonksiyonu, veri setini eğitim (%70) ve test (%30) olarak ikiye böler.
    - `random_state=42`, aynı sonuçları elde etmek için rastgeleliği sabitler.
  3. **Karar Ağacı Modelinin Oluşturulması:**
    - `DecisionTreeClassifier()` ile karar ağacı modeli oluşturulur.
  4. **Modelin Eğitilmesi:**
    - `fit()` fonksiyonu ile model, eğitim verileri kullanılarak eğitilir.
  5. **Tahmin Yapılması:**
    - `predict()` fonksiyonu ile model, test verileri kullanılarak tahminler yapar.
  6. **Doğruluk Skorunun Hesaplanması:**
    - `accuracy_score()` fonksiyonu ile modelin doğruluğu hesaplanır ve yazdırılır.

Bu örnek, denetimli öğrenme (supervised learning) yöntemlerini kullanarak bir sınıflandırma modelinin nasıl oluşturulacağına dair temel bir kılavuz sunar. Karar ağacı algoritması dışında, farklı denetimli öğrenme algoritmaları (örneğin, k-en yakın komşu, destek vektör makineleri, naive Bayes) kullanarak da benzer bir süreç izleyebilirsiniz.