Makine Öğrenmesi

# Nedir

**Geleneksel Bilgisayar Programlaması**

* **Tanım:** Programcıların, bilgisayarın belirli bir görevi yerine getirmesi için açık kurallar ve talimatlar yazdığı yöntemdir.
* **İşleyiş:** Girdi verileri ve programın birleşimiyle çıktı elde edilir. Formül: Girdi + Program = Çıktı.
* **Özellikler:**
  + Kurallar manuel olarak yazılır.
  + Veri bağımlılığı düşüktür.
  + Değişiklikler için kodda manuel güncellemeler gerekir.

**Makine Öğrenmesi (ML)**

* **Tanım:** Bilgisayarların, verilerden öğrenerek ve deneyim kazandıkça performanslarını artırarak kararlar almasını sağlayan bir yapay zeka dalıdır.
* **İşleyiş:** Girdi ve çıktı verileri algoritmalara sunularak bir model oluşturulur. Formül: Girdi + Çıktı = Program.
* **Özellikler:**
  + Model, verilerden kalıpları öğrenir.
  + Veri bağımlılığı yüksektir.
  + Yeni verilere uyum sağlamak için model yeniden eğitilebilir.

**Benzerlikler**

* Her iki yöntem de bilgisayarların belirli görevleri yerine getirmesini sağlar.
* Her ikisi de algoritmalar kullanır.

**Farklılıklar**

* **Kodlama Yaklaşımı:**
  + Geleneksel programlamada kurallar manuel yazılır.
  + Makine öğrenmesinde model, verilerden öğrenir.
* **Veri Bağımlılığı:**
  + Geleneksel programlama, veri bağımlılığı düşüktür.
  + Makine öğrenmesi, veri bağımlılığı yüksektir.
* **Uyarlanabilirlik:**
  + Geleneksel programlama, değişiklikler için manuel güncellemeler gerektirir.
  + Makine öğrenmesi, yeni verilere uyum sağlamak için model yeniden eğitilebilir.

# Makine öğrenmesi tarihçesi

**Perceptron ve Frank Rosenblatt (1958)**

1958’de Frank Rosenblatt, biyolojik nöronlardan ilham alarak **Perceptron** adı verilen ilk yapay sinir ağını geliştirdi. Perceptron, temel olarak bir giriş vektörünü ağırlıklandırarak ve belirli bir eşik değeriyle karşılaştırarak çıktı üreten basit bir öğrenme modeliydi. İlk başta umut vadeden bu model, daha sonra karmaşık problemlerde yetersiz kaldığı için eleştirildi.

**AI Winter ve Minsky-Papert Eleştirileri (1970’ler)**

Marvin Minsky ve Seymour Papert, 1969’da yazdıkları *Perceptrons* kitabında, perceptronların belirli problemlerde başarısız olduğunu gösterdi. Özellikle, XOR gibi doğrusal olarak ayrıştırılamayan problemlerin çözülemeyeceğini belirttiler. Bu eleştiriler, yapay zeka (AI) araştırmalarına olan ilgiyi azalttı ve **AI Winter (Yapay Zeka Kışı)** olarak adlandırılan bir duraklama dönemine neden oldu.

**Machine Learning’in Tekrar Doğuşu (1980’ler - 1990’lar)**

1980’lerde, çok katmanlı sinir ağları (MLP - Multi-Layer Perceptron) ve **geri yayılım (backpropagation)** algoritmasıyla AI tekrar ilgi görmeye başladı. Bu dönemde istatistiksel ve olasılıksal yöntemlerin yapay zekaya entegre edilmesiyle **Makine Öğrenmesi (Machine Learning)** daha bağımsız bir alan olarak gelişti.

**AI, ML ve DL Nedir?**

* **AI (Artificial Intelligence - Yapay Zeka)**: İnsan benzeri bilişsel yetenekleri taklit eden sistemlerin genel adıdır.
* **ML (Machine Learning - Makine Öğrenmesi)**: AI’nin bir alt dalı olup, makinelerin verilerden öğrenmesini sağlayan yöntemleri kapsar.
* **DL (Deep Learning - Derin Öğrenme)**: ML’in bir alt dalı olup, çok katmanlı sinir ağlarıyla büyük miktarda veriden daha karmaşık öğrenme yapabilen modelleri içerir.

**TD-Gammon (1994)**

1994 yılında Gerald Tesauro, **TD-Gammon** adlı bir geriye yayılım sinir ağı modelini kullanarak tavla oynayan bir AI geliştirdi. Model, **takviyeli öğrenme (reinforcement learning)** kullanarak kendi oyun seviyesini artırdı ve dünya çapında insan oyunculara rakip olabilecek bir düzeye ulaştı.

**Deep Blue (1997)**

IBM tarafından geliştirilen **Deep Blue**, 1997’de satrançta dünya şampiyonu Garry Kasparov’u yenerek tarihte bir ilke imza attı. Deep Blue, kurallara dayalı bir sistem olmasına rağmen, AI araştırmalarına büyük ilgi uyandırdı.

**Jeopardy ve IBM Watson (2011)**

IBM'in geliştirdiği **Watson**, 2011’de Jeopardy yarışmasını kazandı. Doğal dil işleme (NLP) ve makine öğrenmesini birleştiren Watson, karmaşık soruları anlamak ve cevaplamak için büyük miktarda veriyi analiz edebiliyordu.

**AlphaGo (2016)**

Google DeepMind tarafından geliştirilen **AlphaGo**, 2016 yılında Go oyununda dünya şampiyonu Lee Sedol’u mağlup etti. AlphaGo, **Derin Öğrenme (Deep Learning) ve Takviyeli Öğrenme (Reinforcement Learning)** kombinasyonuyla çalışıyordu.

**AlphaStar (2019)**

Google DeepMind’in geliştirdiği **AlphaStar**, 2019’da StarCraft II oyununda profesyonel oyunculara karşı galip geldi. AlphaStar, karmaşık stratejileri öğrenmek için **güçlendirmeli öğrenme** ve **gelişmiş sinir ağları** kullandı.

Bu başlıkların dışında eklenmesi gereken bazı konular:

* **Backpropagation (1986)**: Sinir ağlarının öğrenmesini sağlayan önemli bir algoritma.
* **GANs (Generative Adversarial Networks - 2014)**: Üretici yapay zeka modellerinin ortaya çıkışı.
* **Transformer Modelleri (2017)**: Doğal dil işleme alanında büyük ilerleme sağlayan modeller (örneğin, GPT serisi).

# Makine Öğrenmesi Türleri ve Alt Dalları

Makine öğrenmesi, problemleri çözmek için kullanılan veri odaklı yaklaşımlara göre üç ana kategoriye ayrılır:

**1. Denetimli Öğrenme (Supervised Learning)**

Bu yöntemde model, etiketlenmiş verilerle eğitilir. Model, giriş ve çıkış arasındaki ilişkiyi öğrenerek yeni verilere doğru tahminler yapmayı amaçlar.

* **Örnekler:**
  + **Spam Filtreleme:** E-postaların spam olup olmadığını sınıflandırma.
  + **Arama Motorları:** Kullanıcının aradığı kelimeye en uygun sonucu göstermek için denetimli öğrenme kullanılır.
  + **Tıbbi Teşhis:** Hastalık teşhisinde, geçmiş hasta verileriyle eğitim alan sistemler.
* **Alt Dalları:**
  + **Regresyon:** Sürekli verileri tahmin etmek (ör. ev fiyatları tahmini).
  + **Sınıflandırma:** Verileri belirli kategorilere ayırmak (ör. kedi veya köpek).

**2. Denetimsiz Öğrenme (Unsupervised Learning)**

Bu yöntemde model, etiketlenmemiş veriler üzerinde çalışır ve örüntüler bulmaya odaklanır. Modelin amacı, veri içindeki gizli yapıları keşfetmektir.

* **Örnekler:**
  + **Müşteri Segmentasyonu:** E-ticaret sitelerinde müşterileri alışkanlıklarına göre gruplandırmak.
  + **Anomali Tespiti:** Bankalarda sahte işlemleri tespit etmek.
  + **Öneri Sistemleri:** Netflix veya Spotify gibi platformlarda kullanıcıya uygun öneriler sunmak.
* **Alt Dalları:**
  + **Kümeleme (Clustering):** Benzer verileri gruplayarak anlamlı segmentler oluşturmak (ör. müşteri kümeleri).
  + **Boyut Azaltma (Dimensionality Reduction):** Verileri daha az değişkenle temsil etmek (ör. PCA algoritması).

**3. Takviyeli Öğrenme (Reinforcement Learning)**

Bu yöntemde model, ödül-ceza sistemine dayalı olarak çevresiyle etkileşim kurar ve zaman içinde en iyi stratejiyi öğrenir.

* **Örnekler:**
  + **Oyun Yapay Zekaları:** AlphaGo ve AlphaStar gibi sistemler takviyeli öğrenme kullanarak en iyi oyun stratejisini öğrenir.
  + **Otonom Araçlar:** Trafik kurallarına ve çevresel koşullara uygun şekilde araç kullanmayı öğrenmek.
  + **Robotik:** Robotların, belirli görevleri en iyi şekilde yerine getirmesi için eğitilmesi.
* **Alt Dalları:**
  + **Q-Learning:** Modelin zaman içinde en iyi aksiyonları seçmesini sağlayan bir yöntem.
  + **Derin Takviyeli Öğrenme (Deep RL):** Derin sinir ağlarıyla desteklenen takviyeli öğrenme (ör. AlphaGo).

**Makine Öğrenmesi Kullanım Alanları**

* **Arama Motorları:** Google, Bing gibi sistemler, kullanıcının geçmiş aramalarına ve içeriğin alaka düzeyine göre sonuçları sıralamak için makine öğrenmesi kullanır.
* **Sosyal Medya:** Facebook, Instagram gibi platformlar, kullanıcı etkileşimlerini analiz ederek kişiselleştirilmiş içerikler sunar.
* **Finans:** Bankalar, kredi skorlama, dolandırıcılık tespiti ve yatırım tahminleri için ML kullanır.
* **Sağlık:** MR görüntülerinden kanser teşhisi, ilaç keşfi gibi alanlarda yapay zeka destekli modeller geliştirilir.

# Makine Öğrenmesi mi, Derin Öğrenme mi?

Makine öğrenmesi (ML) ve derin öğrenme (DL) sıkça karıştırılan ancak farklı yaklaşımlar sunan iki alandır. Peki, hangisini öğrenmek daha mantıklı? Bu soruya cevap vermek için temel farkları ve kullanım alanlarını anlamak gerekir.

**Makine Öğrenmesi (ML) vs. Derin Öğrenme (DL)**

Makine öğrenmesi, genel olarak istatistiksel yöntemler ve algoritmalar kullanarak verilerden öğrenmeyi sağlar. Derin öğrenme ise makine öğrenmesinin bir alt dalıdır ve çok katmanlı yapay sinir ağlarını (deep neural networks) kullanarak daha karmaşık ilişkileri öğrenebilir.

Aşağıdaki tabloda temel farkları özetleyelim:

| **Kriter** | **Makine Öğrenmesi (ML)** | **Derin Öğrenme (DL)** |
| --- | --- | --- |
| **Veri İşleme** | Özellik mühendisliği (**feature extraction**) gerektirir. | Özellikler otomatik olarak öğrenilir. |
| **Öğrenme Yapısı** | Karar ağaçları, destek vektör makineleri, lojistik regresyon gibi algoritmalar kullanılır. | Sinir ağları kullanılır (CNN, RNN, Transformer vb.). |
| **Veri Miktarı** | Az ve orta miktarda verilerle iyi çalışır. | Büyük veri setlerine ihtiyaç duyar. |
| **Donanım Gereksinimi** | CPU üzerinde çalışabilir. | Yüksek performanslı GPU’lar gerektirir. |
| **Eğitim Süresi** | Daha hızlıdır. | Daha uzun sürebilir. |
| **Örnek Kullanım** | Kümeleme, tahminleme, öneri sistemleri. | Görüntü işleme, doğal dil işleme, otonom araçlar. |

**Hangi Durumda Hangisi Seçilmeli?**

* **Veriniz küçükse ve hızlı çözümler üretmek istiyorsanız → Makine Öğrenmesi**
* **Büyük veri ve karmaşık problemleriniz varsa → Derin Öğrenme**
* **Özellik mühendisliği yapmak istiyorsanız → Makine Öğrenmesi**
* **Özellik mühendisliğiyle uğraşmadan modelin otomatik öğrenmesini istiyorsanız → Derin Öğrenme**

Eğer temel düzeyde başlayıp ilerlemek istiyorsanız **önce makine öğrenmesini öğrenmek**, ardından derin öğrenmeye geçmek daha mantıklıdır. Çünkü makine öğrenmesi, veriyi anlama ve modelleme süreçlerini daha iyi kavramanızı sağlar.

**ML ve DL İşleyiş Şemaları Ne Anlama Geliyor?**

**Makine Öğrenmesi İşleyişi (ML)**

📌 **Input → Feature Extraction → Classification → Output**

* **Feature Extraction (Özellik Çıkarma):** Veriden anlamlı özellikler seçilir (örneğin, bir fotoğrafta kenar tespiti yapmak).
* **Classification (Sınıflandırma):** Seçilen özellikler, bir model (örneğin karar ağaçları veya lojistik regresyon) kullanılarak sınıflandırılır.
* **Örnek:** Bir el yazısı rakam tanıma sisteminde, ML önce rakamın kenarlarını ve şekillerini analiz eder, ardından sınıflandırma modeliyle hangi rakam olduğunu tahmin eder.

**Derin Öğrenme İşleyişi (DL)**

📌 **Input → Feature Extraction + Classification → Output**

* **Feature Extraction ve Classification aynı anda yapılır:** Derin öğrenme modelleri, özellikleri insan müdahalesi olmadan otomatik çıkarır.
* **Örnek:** Bir derin sinir ağı, el yazısı rakamları tanırken kenar tespitini, şekil tanımayı ve sınıflandırmayı kendi içinde yapar. Özellik mühendisliğine gerek kalmaz.

Kısacası, **ML insan eliyle seçilen özellikleri kullanırken, DL bu özellikleri kendisi öğrenir.**

**Sık Sorulan Sorular**

❓ **Makine öğrenmesini bilmeden derin öğrenme öğrenilir mi?**  
Evet, ama önerilmez. Makine öğrenmesi, veri analizi ve modelleme sürecini anlamanıza yardımcı olur. ML öğrenmeden DL’ye geçenler, genellikle modelin neden başarısız olduğunu anlamakta zorlanır.

❓ **Derin öğrenme her zaman daha mı iyidir?**  
Hayır. Küçük veri setlerinde ve düşük donanım gereksinimi olan durumlarda klasik makine öğrenmesi genellikle daha etkilidir.

❓ **Makine öğrenmesi mi, yapay zeka mı?**  
Makine öğrenmesi, yapay zekanın bir alt dalıdır. Yapay zeka, ML ve DL’yi kapsayan daha geniş bir alanı ifade eder.

❓ **Derin öğrenme neden GPU gerektirir?**  
Çünkü çok fazla matris işlemi içerir ve GPU’lar bu işlemleri paralel olarak hızlandırabilir.

# Makine Öğrenmesine Giriş: Temel Kavramlar ve Veri ile Tanışma

Makine öğrenmesi, temel olarak **veriyi anlama ve anlamlandırma** sürecidir. Bu sürecin sonucunda elde edilen bilgiye **"öğrenme" (learning)** denir. Makine öğrenmesi, öğrenme biçimlerine göre iki temel kategoriye ayrılır:

**1. Gözetimli Öğrenme (Supervised Learning)**

Gözetimli öğrenmede, **girdi-çıktı ilişkisi** belirli olan veriler kullanılır. Başlangıçta her girdinin hangi çıktıya karşılık geldiği bilinir.

* Amaç, verilen **etiketli veriler** üzerinden yeni girdiler için doğru çıktıyı tahmin etmektir.
* Model, geçmiş verilerden öğrenerek **genelleme yapmaya** çalışır.
* Örnekler:
  + E-posta sınıflandırması (*spam* veya *spam değil*)
  + Hastalık teşhisi (Bir hastanın hasta olup olmadığı)
  + Ev fiyat tahmini (Ev özelliklerine göre fiyat belirleme)

Gözetimli öğrenme algoritmaları genellikle **regresyon** ve **sınıflandırma** olmak üzere ikiye ayrılır:

* **Regresyon (Regression):** Sürekli değerleri tahmin eder (Örneğin, bir evin fiyatı)
* **Sınıflandırma (Classification):** Belirli kategorilere ayırır (Örneğin, e-postaların spam olup olmaması)

**2. Gözetimsiz Öğrenme (Unsupervised Learning)**

Gözetimsiz öğrenmede, elimizde **yalnızca girdiler** vardır; ancak bu girdilere karşılık gelen etiketli çıktılar mevcut değildir.

* Amaç, **veri içindeki desenleri ve ilişkileri** keşfetmektir.
* Model, veriler arasındaki **benzerlikleri** bularak yapısal örüntüler oluşturur.
* Örnekler:
  + Müşteri segmentasyonu (Benzer alışveriş alışkanlıklarına sahip müşterileri gruplama)
  + Anomali tespiti (Finansal dolandırıcılık belirleme)
  + Belge kümeleme (Benzer içerikli belgeleri gruplama)

Gözetimsiz öğrenme genellikle şu iki temel yönteme dayanır:

* **Kümeleme (Clustering):** Veriyi **benzer özelliklere göre** gruplandırır (Örneğin, k-means algoritması)
* **Boyut indirgeme (Dimensionality Reduction):** Verideki **önemli özellikleri** belirleyerek boyutları azaltır (Örneğin, PCA - Principal Component Analysis)

# Notasyon - Yazım Kurallarımız

Kursumuz boyunca kullanacağımız temel notasyon şu şekilde olacaktır:

**n:**  
Elimizdeki veri setinin her bir gözlem değeri.  
Satır Sayısı aslında.  
Örneğin, Maaş verilerini tutan Wage.csv'deki her bir satır bir erkeğe ait maaş verisini tutuyordu.  
İşte toplam satır sayısına n diyeceğiz.

**p:**  
Değişken sayısı.  
Veri içindeki her bir kriter yani Sütun Sayısıdır.  
Örneğin, Maaş verilerindeki sütunlar: year, age, sex, maritl, race, education ...

**x\_ij:**  
i'inci satır, j'inci sütundaki değer.  
Yani, i'inci gözlem değeri için j'inci değişken.

**X:**  
Tüm verileri tutan girdi (input) matrisi.  
Notasyon: Girdi Matrisi

**y:**  
Her bir gözlem (veri satırı) değerinin sonucu.  
Label olarak geçer.  
Her satırın aslında gerçekte ne olduğunu tuttuğumuz değişken.  
Notasyon: Çıktı Vektörü

Verimiz aslında şu şekildedir:  
(x1, y1), (x2, y2), (x3, y3), ..., (xn, yn)

Yani her bir satır -> (gözlem, label)