

Makine Öğrenmesi

Yapay Zeka ve Uygulamaları - Copyright © 2026

Makine öğreniminin kökenleri, "Makineler düşünebilir mi?" sorusunu ortaya atan Alan Turing'e dayanır.

Yapay öğrenme veya makine öğrenme, insan öğrenme yeteneklerini taklit edebilen her türlü öğrenmeyi ifade eder ve yapay zekânın bir alanı olarak geniş bir uygulama alanına sahiptir.



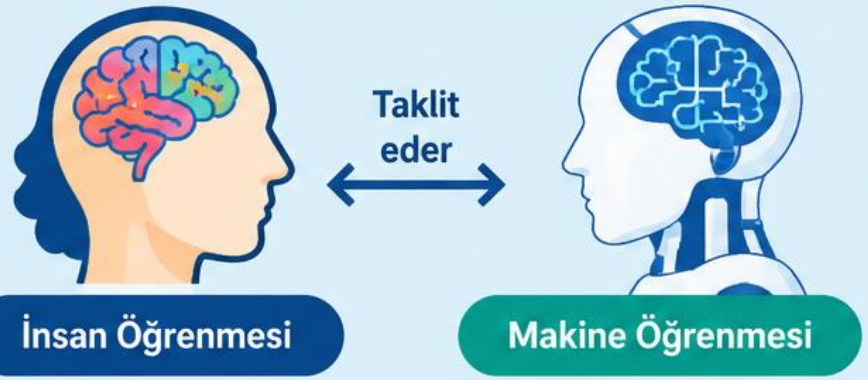
Makine öğreniminin kökenleri, **"Makineler düşünebilir mi?"** sorusunu ortaya atan **Alan Turing**'e dayanır.

Makineler düşünebilir mi?



Yapay Öğrenme / Makine Öğrenme

İnsan öğrenme yeteneklerini taklit edebilen her türlü öğrenmeyi ifade eder ve **yapay zekânın** bir alanı olarak geniş bir uygulama alanına sahiptir.



Geniş Uygulama Alanı

- **Doğal Dil İşleme** (Sohbet robotları, çeviri)
- **Veri Analizi** (Öngörüler, karar destek)
- **Görüntü İşleme** (Resim tanıma, yüz tanıma)
- **Otonom Sistemler** (Otonom araçlar, robotlar)
- **Sağlık** (Erken teşhis, kişiselleştirilmiş tedavi)
- **Eğitim** (Uyarlanabilir öğrenme, akıllı sistemler)

Yapay Zeka ve Uygulamaları - Copyright © 2026

Makine öğrenimi, bilgisayar sistemlerinin talimatlar yerine düzenlere ve çıkarıma dayalı olarak görevleri yerine getirmesi için algoritmalar ve istatistiksel modeller geliştiren bir disiplindir.

Bu algoritmalar, büyük miktarda geçmiş veriyi analiz ede-rek ve veri düzenlerini tanımlayarak çalışır. Bu şekilde, belirli bir giriş veri küme-sinden daha doğru sonuçlar elde edebilirler.

Yapay Zeka ve Uygulamaları - Copyright © 2026



Makine öğrenimi, bilgisayar sistemlerinin talimatlar yerine düzenlere ve çıkarıma dayalı olarak görevleri yerine getirmesi için **algoritmalar** ve **istatistiksel modeller** geliştiren bir disiplindir. Bu algoritmalar, **büyük miktarda geçmiş veriyi analiz ederek** ve veri düzenlerini tanımlayarak çalışır. Bu şekilde, belirli bir giriş veri kümesinden daha doğru sonuçlar elde edebilirler.

1. Veri Toplama

Geçmiş veriler, çeşitli kaynaklardan toplanır ve bir veri kümesi oluşturulur.

2. Analiz ve Öğrenme

Algoritmalar, verileri analiz ederek veri düzenlerini ve örüntüleri tanımlar.

3. Model Geliştirme

Bulunan düzenlere göre istatistiksel modeller ve algoritmalar oluşturulur.

4. Tahmin / Karar

Model, yeni giriş verilerini kullanarak daha doğru sonuçlar üretir.

5. Daha İyi Sonuçlar

Sürekli öğrenme ve gelişim sayesinde doğruluk oranı artar.

Temel Özellikleri

Algoritmalar
Verilerden öğrenmeyi sağlayan adım adım kurallar.

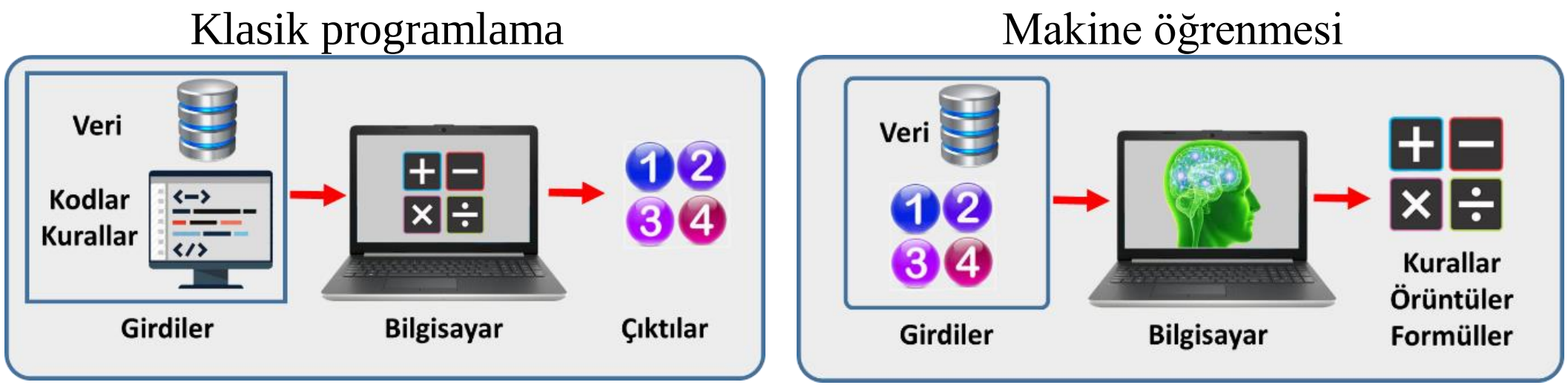
İstatistiksel Modeller
Olasılık ve istatistik temelli yaklaşımlar.

Büyük Veri
Geçmiş verilerin analiz edilmesi.

Daha Doğru Tahminler
Veri düzenlerini tanıyarak daha güvenilir sonuçlar üretme.

Yapay Zeka ve Uygulamaları - Copyright © 2026

Örneğin, etiketlenmiş (sınıflandırılmış) fotoğraflardan oluşan bir veri seti kullanılarak bir makine öğrenme algoritması eğitilebilir. Bu algoritma, fotoğraflardan sınıfları betimleyen özellikleri çıkarır ve yeni bir fotoğrafın bu özelliklere ne kadar uyduğunu değerlendirerek sınıflandırma yapar.



Şekil 7.1. Klasik programlama ve Makine Öğrenmesi

Makine öğrenmesini klasik programlamayla karşılaştırmak gerekirse, klasik programlamada programcılar tüm olası durumları hesaba katarak koşullu ifadeler yazarlar ve programlar bu kodlar doğrultusunda işlem yapar. Ancak makine öğrenmesinde, veri ve çıktılar üzerinde çalışarak olası ilişkiler, bağlantılar, şablonlar ve örüntüler ortaya çıkarılır.

Bu sayede, her veri seti için yeni kod yazma ihtiyacı ortadan kalkar ve daha esnek ve genel geçerli çözümler elde edilir.

Yapay Zeka ve Uygulamaları - Copyright © 2026

Klasik Programlama

Tüm olası durumları kodla, program çalışsın.



```
if (yas >= 18) {
   console.log("Yetişkin");
} else if (yas >= 13) {
   console.log("Genç");
} else {
   console.log("Çocuk");
}
```



Programcı

Koşullu ifadeler ve kurallar

Sonuç

Yapay Zeka ve Uygulamaları - Copyright © 2026

Makine Öğrenmesi

Veriden öğren, örüntüleri keşfet, çıktı üret.



Klasik Programlama

Tüm olası durumları kodla, program çalışsın.

Nasıl Çalışır?

- Programcı, tüm olası durumları hesaba katar.
- Koşullu ifadeler, döngüler ve mantık kuralları yazar.
- Program, bu kodlar doğrultusunda işlem yapar.

Dezavantajı

- Her yeni veri seti veya senaryo için yeni kod yazmak gerekir.
- Kurallar ne kadar kapsamlı yazılırsa yazılsın, bazı durumlar gözden kaçabilir.
- Esnekliği sınırlıdır.

Makine Öğrenmesi

Veriden öğren, örüntüleri keşfet, çıktı üret.

Nasıl Çalışır?

- Geçmiş veriler analiz edilir.
- Verilerdeki ilişkiler, bağlantılar, şablonlar ve örüntüler bulunur.
- Bu öğrenme sonucunda, her veri seti için yeniden kod yazmaya gerek kalmadan, daha esnek ve **genel geçerli çözümler** üretilir.

Avantajları

- Yeni veri setleri için kod yazma ihtiyacı ortadan kalkar.
- Değişen durumlara uyum sağlayabilir.
- Daha esnek, genelleştirilebilir ve gelişime açıktır.

Klasik programlama, kurallarla çalışır.
Makine öğrenmesi, verilerden öğrenir.

Klasik programlama ile gerçekleştirilebilecek karmaşık işlemler, makine öğrenmesi yaklaşımıyla daha kolay ve etkili bir şekilde gerçekleştirilebilir. Makine öğrenimi, girdi ve çıktılar arasındaki ilişkileri belirleyerek veri analizi ve tahmin yeteneklerini artırırken, klasik programlama yöntemlerinden farklı olarak daha geniş ve dinamik bir çözüm yelpazesi sunar.

Makine Öğrenimi (ML), bilgisayar sistemlerinin büyük hacimli verilerden öğrenmesini ve belirli sorunları çözmesini sağlayan bir Yapay Zeka uygulamasıdır. Bu algoritmalar deneyim yoluyla verimliliklerini otomatik olarak artırır [Menon, 2023]

Yapay Zeka ve Uygulamaları

Klasik programlama ile gerçekleştirilebilecek karmaşık işlemler, makine öğrenmesi yaklaşımıyla daha kolay ve etkili bir şekilde gerçekleştirilebilir. Makine öğrenimi, girdi ve çıktılar arasındaki ilişkileri belirleyerek veri analizi ve tahmin yeteneklerini artırırken, klasik programlama yöntemlerinden farklı olarak daha geniş ve dinamik bir çözüm yelpazesi sunar.

Klasik Programlama

Belirli kurallara göre, adım adım işlemlerle çözüm üretir.

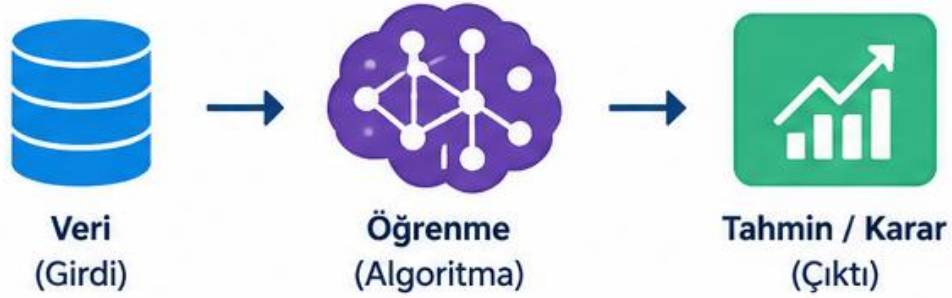


- Önceden tanımlı kurallara dayanır.
- Her adım açıkça belirlenir.
- Yeni durumlara uyum sağlamak için kod değişikliği gerekir.
- Karmaşık ve büyük veri setlerinde sınırlı kalabilir.

Daha Esnek
Daha Güçlü
Çözümler

Makine Öğrenmesi

Verilerden öğrenerek, girdi ve çıktı arasındaki ilişkileri keşfeder.



- Verilerden örüntüleri öğrenir.
- Deneyim yoluyla performansını otomatik olarak artırır.
- Girdi ve çıktı arasındaki ilişkileri analiz eder.
- Daha geniş ve dinamik bir çözüm yelpazesi sunar.

Öncelikle makine öğrenimi, bilgisayar biliminin bir alt alanı olarak büyüyen yenilikçi bir teknolojidir.

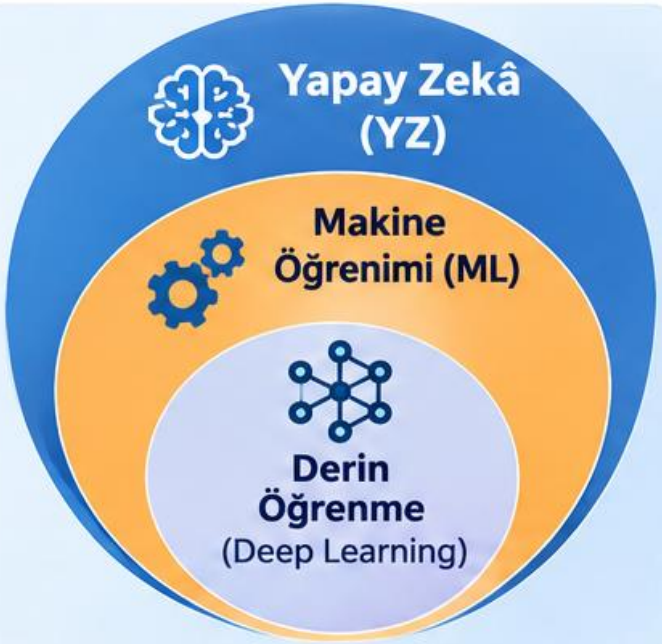
Makine öğrenimi algoritmaları verilerden öğrenir ve öğrenme sonuçlarını insanların bilinçli kararlar almasına yardımcı olmak için kullanır.

Makine öğrenimi bağlamında "öğrenme" kelimesi, makinelerin verileri analiz etme ve bu analizden bazı sonuçlar üretme yeteneğini ifade eder [Danielkievich, 2023].

Makine Öğrenimi (ML), Yapay Zekanın bir alt alanı olarak düşünülebilir çünkü bu algoritmalar, bilgisayarların sadece veri öğelerini depolamak ve almak gibi basit işlevler yerine, verilerden genelleme yaparak daha akıllıca davranmayı öğrenmelerini sağlayan yapı taşları olarak görülebilir. Bu alan, bilgisayar bilimi, istatistik, biyoloji ve psikoloji gibi çeşitli akademik disiplinlerden ilham almıştır. Makine öğrenimi girişimlerinin temel işlevi, bilgisayarlara geçmiş deneyimlere dayanarak iyi bir tahminciyi otomatik olarak nasıl bulacaklarını belirlemektir ve bu iş iyi bir sınıflandırıcı tarafından yapılır [Muhammad, Yan, 2015].

Makine Öğrenimi (ML) Nedir?

Makine Öğrenimi (ML), Yapay Zekânın (YZ) bir alt alanıdır. Bilgisayarların, sadece verileri depolamak ve almak gibi basit işlevler yerine, verilerden genelleme yaparak daha akıllıca davranmayı öğrenmelerini sağlayan algoritmalar bütünüdür.



Çeşitli Disiplinlerden İlham Alır

Bilgisayar Bilimi
(Algoritma, veri yapıları, modelleme)

İstatistik
(Veri analizi, olasılık, çıkarım)

Biyoloji
(Öğrenme, uyum, evrim)

Psikoloji
(Öğrenme süreçleri, davranış modelleme)

Temel İşlevi

Bilgisayarlara, geçmiş deneyimlere (verilere) dayanarak iyi bir tahminciyi otomatik olarak nasıl bulacaklarını belirlemektir. Bu işlev çoğunlukla iyi bir sınıflandırıcı (classifier) tarafından gerçekleştirilir [Muhammad, Yan, 2015].



Örnek: Sınıflandırma (Classification)

Verilen bir girdinin hangi kategoriye ait olduğunu öğrenerek tahmin eder.



Girdi
(Çiçek görseli)

→



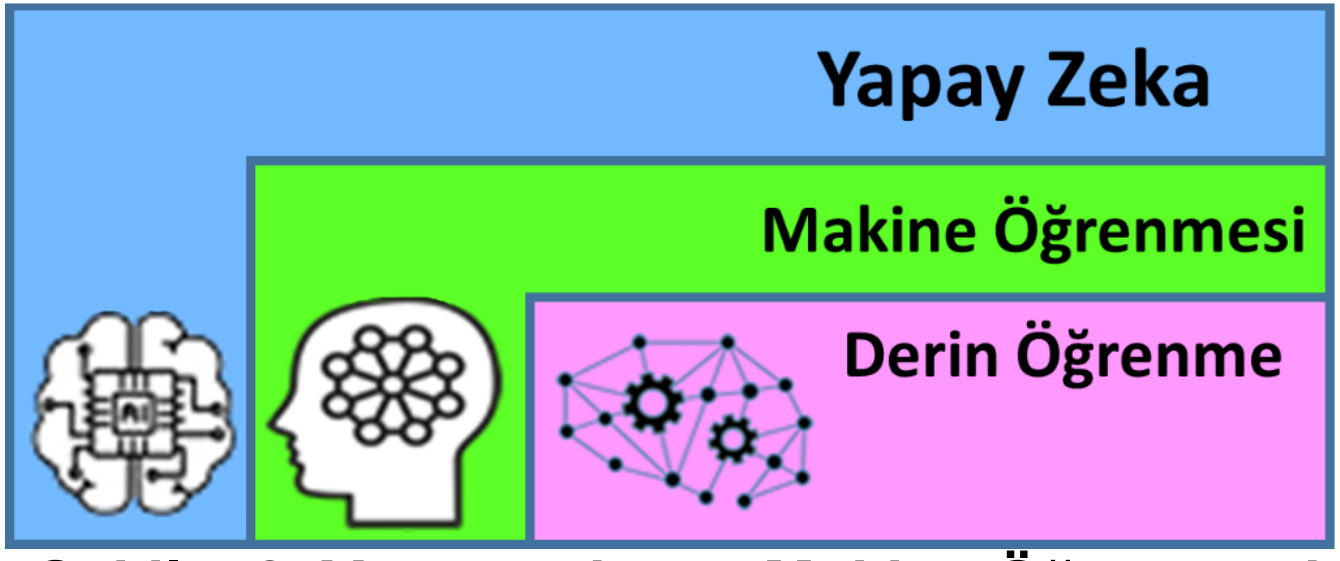
Sınıflandırıcı
(Öğrenilmiş model)

→

Çıktı
"Papatya"
(sınıf etiketi)

Yapay Zeka ve Uygulamaları - Copyright © 2026

Makine öğrenimi, farklı görevleri gerçekleştirmek için bilgisayarlara farklı algoritmalar öğretmek ve makineleri kendi başlarının çaresine bakabilecek hale getirmekle ilgilenen bir teknolojidir. Makine Öğrenimi (ML), belirli bir görevde geliş-meyi sürdürmek için bilgisayar programlarının ve algoritmaların öğretilmesiyle ilgilenen bir çalışma alanıdır. Makineler verilerden çıkarılan içgörülerden yararlanır [PathakJan, 2021].



Şekil 7.2. Yapay zeka ve Makine Öğrenmesi

Makinelerin görevlerin çoğunu yerine getirdiği günümüz dünyasında, işlerin nasıl yapıldığını anlamak ve geleceği öngörebilmek önemlidir. İşte tam da burada maki-ne öğren,0imi devreye giriyor. Makine öğrenimi, makinelerin kendi başlarına öğrenme kabiliyetini kazanmasını ve geçmiş deneyimlere dayanarak tahminlerde bulunmasını sağlar [PathakJan, 2021].

Makine öğrenimi, örnek verileri veya geçmiş deneyimleri kullanarak bir performans kriterini optimize etmek için bilgisayarları programlamaktır. Bazı parametrelere kadar tanımlanmış bir modelimiz vardır ve öğrenme, eğitim verilerini veya geçmiş deneyimleri kullanarak modelin parametrelerini optimize etmek için bir bilgisayar programının yürütülmesidir. Model, gelecekte tahminler yapmak için öngörücü veya verilerden bilgi edinmek için açıklayıcı ya da her ikisi birden olabilir [Alpaydin, 2010].

Makine öğrenimi sadece görüş, konuşma tanıma ve robotik gibi birçok alanda sorunlara çözüm sunmakla kalmaz, aynı zamanda yüz tanıma gibi günlük hayatta kolaylıkla gerçekleştirdiğimiz yeteneklerin temelini oluşturur. Farklı pozlar, ışıklandırma ve saç stilleri gibi değişkenlere rağmen, aile üyelerimizi ve arkadaşlarımızı yüzlerinden tanımamızı sağlar. Ancak bunu bilinçli bir şekilde yapmayız ve bu yeteneğimizi nasıl geliştirdiğimizi açıklayamayız [Alpaydin, 2010].

Makine öğrenimi ayrıca, bir yüz görüntüsünün sadece rastgele piksellerden oluşan bir fotoğraf olmadığını anlamamıza yardımcı olur. Her yüzün belirli yerlerinde bulunan gözler, burun ve ağız gibi yapısal özellikleri vardır. Her insanın yüzü, belirli bir kombinasyondan oluşan bir modeldir. Bir öğrenme programı, bir kişinin yüz örneklerini analiz ederek o kişiye özgü örüntüyü belirler ve daha sonra belirli bir görüntüde bu örüntüyü tanır. Bu, örüntü tanımanın bir örneğidir [Alpaydin, 2010].

Makine öğrenimi, önceden belirlenmiş modellere güvenmek yerine doğrudan verilerden öğrenmek için hesaplama yöntemlerini kullanan algoritmaları içeren bir ya-pay zeka alanıdır (Kanade, 2022). Makine öğrenimi ayrıca, otomatik eğitim tekniklerini kullanarak daha iyi algoritmalar geliştiren genel bir terimdir [Lawton, 2023].

Arthur Samuel tarafından 1959'da ortaya atılan "makine öğrenimi" terimi, bilgisayarların açıkça programlanmadan öğrenme yeteneğini tanımlar [Wakefield, 2024]. Bu, makinelere verilerden ve geçmiş deneyimlerden otomatik olarak öğrenme yeteneği kazandıran bir yapay zeka disiplinidir [Kanade, 2022].

Yapay Zeka ve Uygulamaları - Copyright © 2026

Günümüzde, makine öğrenimi bankacılık, sigortacılık, sağlık hizmetleri ve akıllı cihazlar gibi çeşitli sektörlerde kullanılan popüler bir araçtır [Biba, 2023].

Makine öğrenimi, verilerden öğrenme yeteneği sağlayan algoritmaların ve istatistiksel modellerin geliştirilmesini içerir [Pardeshi, 2023].

Denetimli, denetimsiz ve takviyeli öğrenme gibi çeşitli yöntemleri kapsar ve insan müdahalesi olmadan verilerden tahminlerde bulunmayı ve kararlar almayı amaçlar [Pardeshi, 2023].

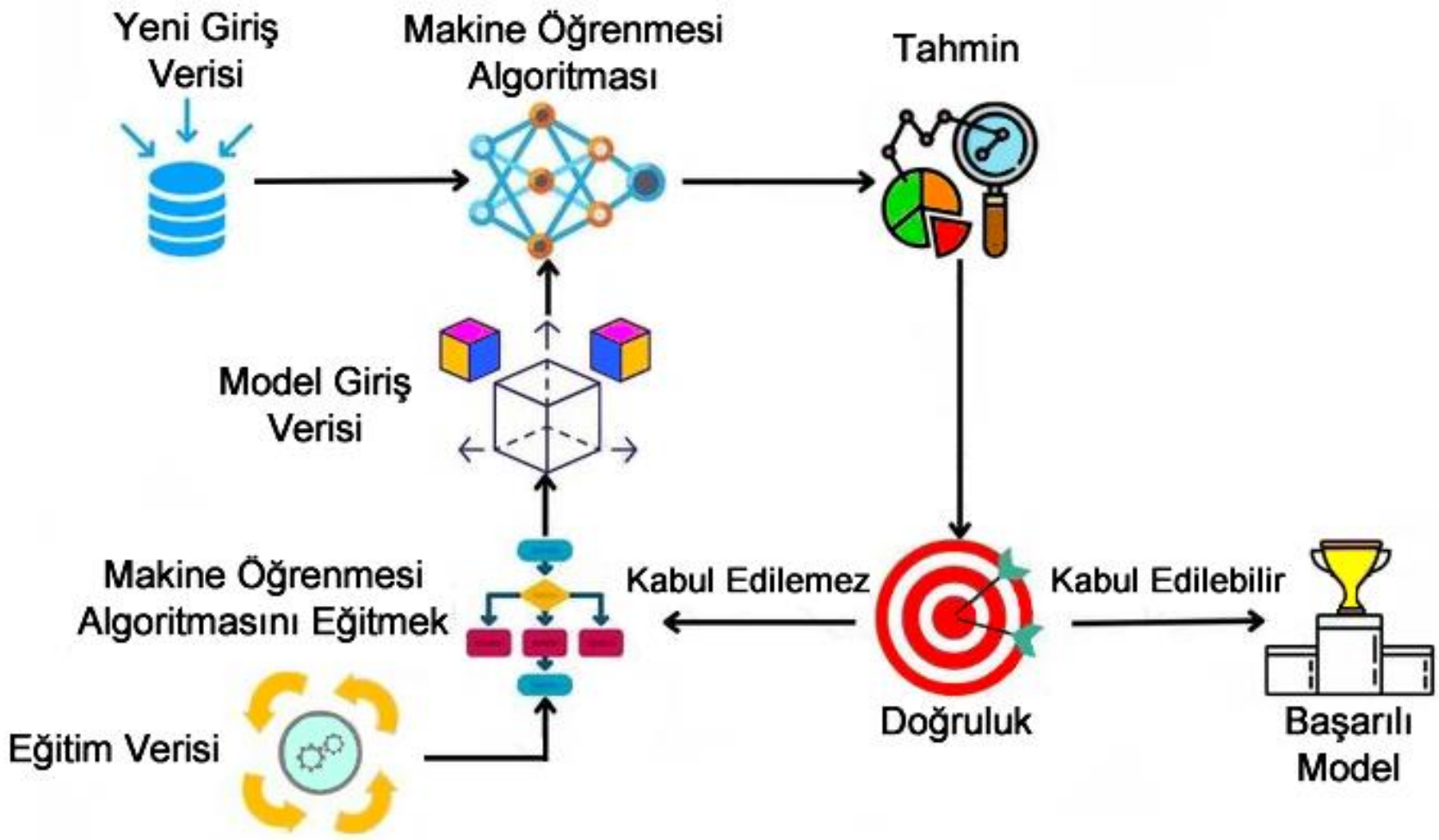
Bu bağlamda, makine öğrenimi bilgisayarların programlanmadan öğrenme yeteneğini geliştiren ve verilerdeki kalıpları tanımlamak için algoritmaları kullanan önemli bir disiplindir.

Denetimli, denetimsiz ve takviyeli öğrenme gibi farklı yöntemlerle çalışarak, veri odaklı tahmin ve kararlar almamıza olanak sağlar.

7.1. Makine Öğrenmesi Çalışma Süreci

Makine öğrenmesi çalışması süreci genellikle belirli adımları takip eden bir süreçtir.

Bu süreç, veri toplama, veri ön işleme, model seçimi, model eğitimi, model değerlendirmesi ve son olarak modelin dağıtımı veya uygulanması aşamalarını içerir.



Şekil 7.3. Makine öğrenmesi çalışma süreci

Yapay Zeka ve Uygulamaları - Copyright © 2026

Şekil 3'te görüldüğü gibi, Makine öğrenmesi çalışma sürecinde, genellikle yeni giriş verileri edinerek başlarız; bu veriler modelimizi eğitmek için temel oluşturur. Yeni giriş verileri ön işlemden geçirilir ve biçimlendirilir, böylece eğitime uygun hale gelir. Ardından, elimizdeki görevin doğasına bağlı olarak uygun bir makine öğrenmesi algoritması seçeriz; bu algoritma, tahminler yapmak için başarılı bir model geliştirmeyi amaçlar. Makine öğrenmesi algoritması, model giriş verilerinden elde edilen eğitim verileriyle eğitilir. Eğitim sırasında, algoritmanın performansını doğruluk metrikleri kullanarak değerlendiririz ve modelin tahminlerinin kabul edilebilir veya kabul edilemez olduğunu belirlemeye çalışırız. Hedefimiz, başarılı bir modeli gösteren bir doğruluk seviyesine ulaşmaktır. Eğitildikten sonra model, yeni veya görünmeyen veriler üzerinde tahminler yapmak için kullanılır ve bu da makine öğrenmesi sürecimizin döngüsünü tamamlar. Bu tekrarlayıcı yaklaşım, modelin geliştirilmesine yardımcı olur ve gerçek dünya uygulamalarında iyi performans gösteren ve güvenilir tahminler üreten bir model elde etmemizi sağlar.

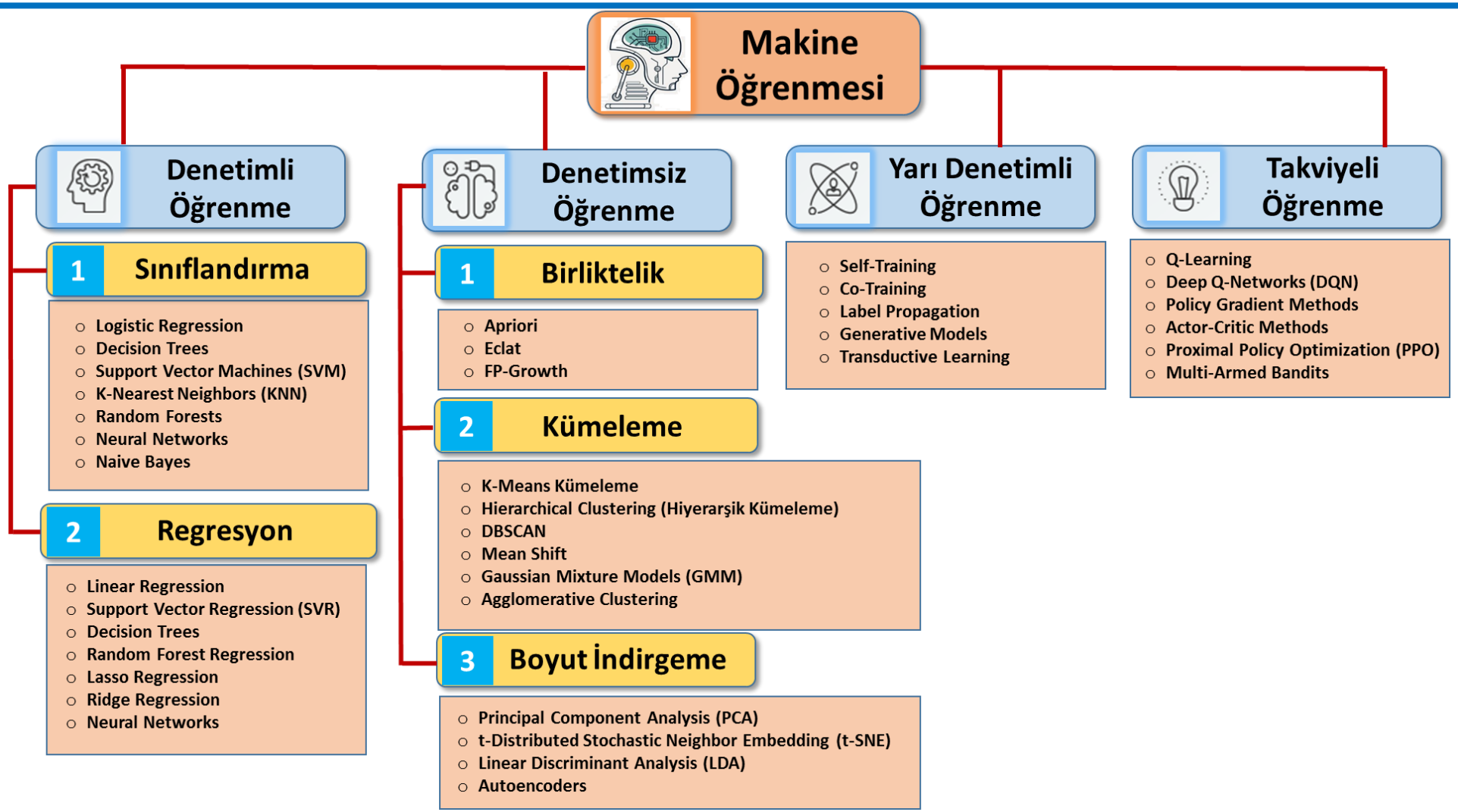
Yapay Zeka ve Uygulamaları - Copyright © 2026

Makine öğrenmesi süreci, veri analizi, istatistik, matematik ve bilgisayar bilimlerinin birleşimiyle desteklenir.

Bu süreçte doğru veri seçimi, uygun model seçimi ve dikkatli değerlendirme, başarılı bir makine öğrenmesi uygulaması için önemlidir.

7.2. Makine Öğrenmesi Türleri

Makine öğrenmesi çalışması süreci genellikle belirli adımları takip eden bir süreç-tir. Bu süreç, veri toplama, veri ön işleme, model seçimi, model eğitimi, model değerlendirmesi ve son olarak modelin dağıtımı veya uygulanması aşamalarını içerir.



Şekil 7.4. Makine öğrenmesi türleri

Makine öğrenimi algoritmaları, her yöntemin artıları ve eksileri olan birçok farklı yöntemle eğitilebilir. Bu yöntemlere ve öğrenme yollarına dayanarak, makine öğrenimi genel olarak dört ana türe ayrılmaktadır [Kanade, 2022].

Makine öğrenimi, çıktı değerlerini kabul edilebilir bir aralıkta tahmin etmek için girdi verilerini alan ve analiz eden programlanmış algoritmaları kullanır. Bu algoritmalara yeni veriler beslendikçe, performansı artırmak için işlemlerini öğrenir ve zaman içinde 'zeka' geliştirir [Wakefield, 2024]. Makine öğrenimi türleri kısaca şunlardır:

Denetimli Öğrenme

Denetimli öğrenme, **etiketli verilerle** eğitilen algoritmalardır. Algoritma, belirli bir giriş değişkeni için doğru çıktıyı bildiği veri seti üzerinde eğitilir. Örneğin, e-postaların spam olup olmadığını tahmin etmek için kullanılabilir.

Nasıl Çalışır?



Etiketli Veri
(Giriş + Çıktı)

Algoritma
(Model Eğitimi)

Tahmin
(Yeni Veri İçin)

Yapay Zeka ve Uygulamaları - Copyright © 2026

Denetimli Öğrenme

Örnek: E-posta Spam Tespiti



Temel Özellikleri

- Etiketli (giriş-çıktı) veriler kullanılır.
- Sınıflandırma ve regresyon gibi görevlerde etkilidir.
- Örnek: Spam tespiti, fiyat tahmini, tıbbi teşhis.

Denetimsiz Öğrenme

Denetimsiz öğrenme, **etiketlenmemiş verilerle** eğitilen algoritmalardır. Algoritma, veriden kalıpları veya ilişkileri kendi başına öğrenmelidir. Örneğin, müşteri davranışlarına göre otomatik olarak müşteri segmentleri oluşturmak için kullanılabilir.

Nasıl Çalışır?



Denetimsiz Öğrenme

Örnek: Müşteri Segmentasyonu



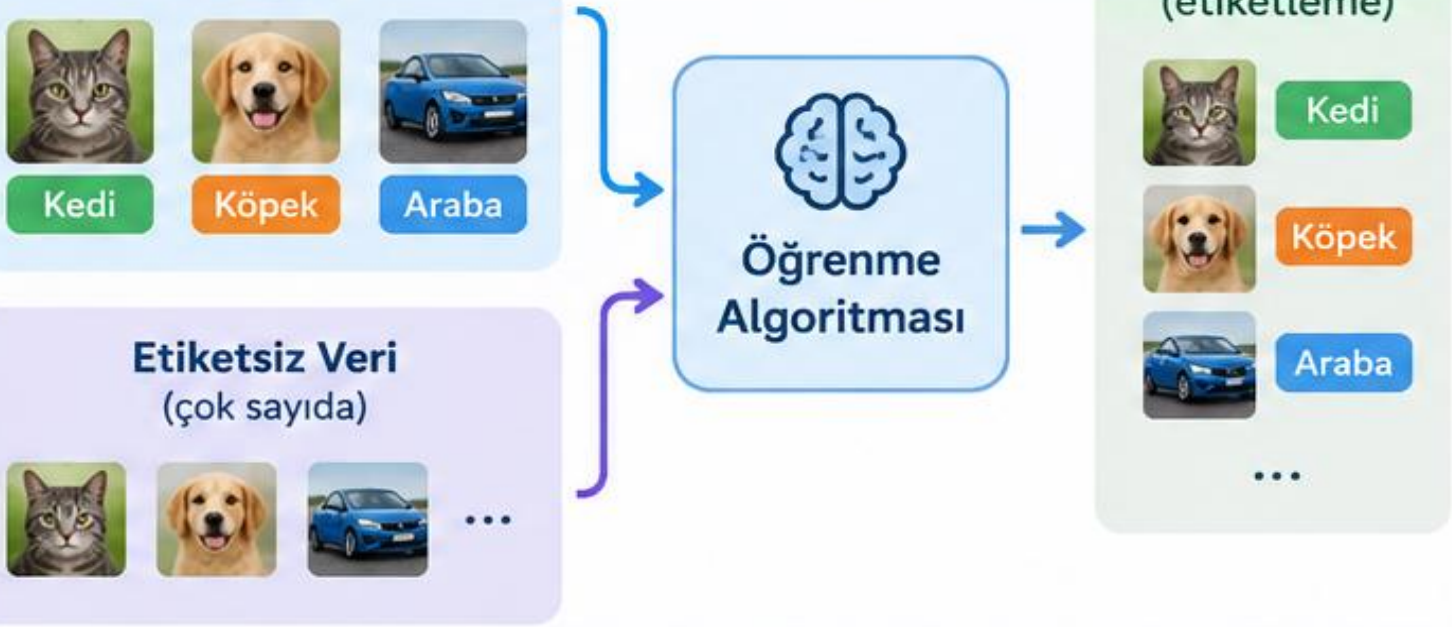
Temel Özellikleri

- Etiketlenmemiş veriler kullanılır.
- Kalıp, grup ve ilişki keşfi yapılır.
- Örnek: Müşteri segmentasyonu, özellik indirgeme, anomali tespiti.

Yapay Zeka ve Uygulamaları - Copyright © 2026

Yarı Denetimli Öğrenme

Hem etiketli hem de etiketsiz verilerin kullanıldığı öğrenme türüdür.



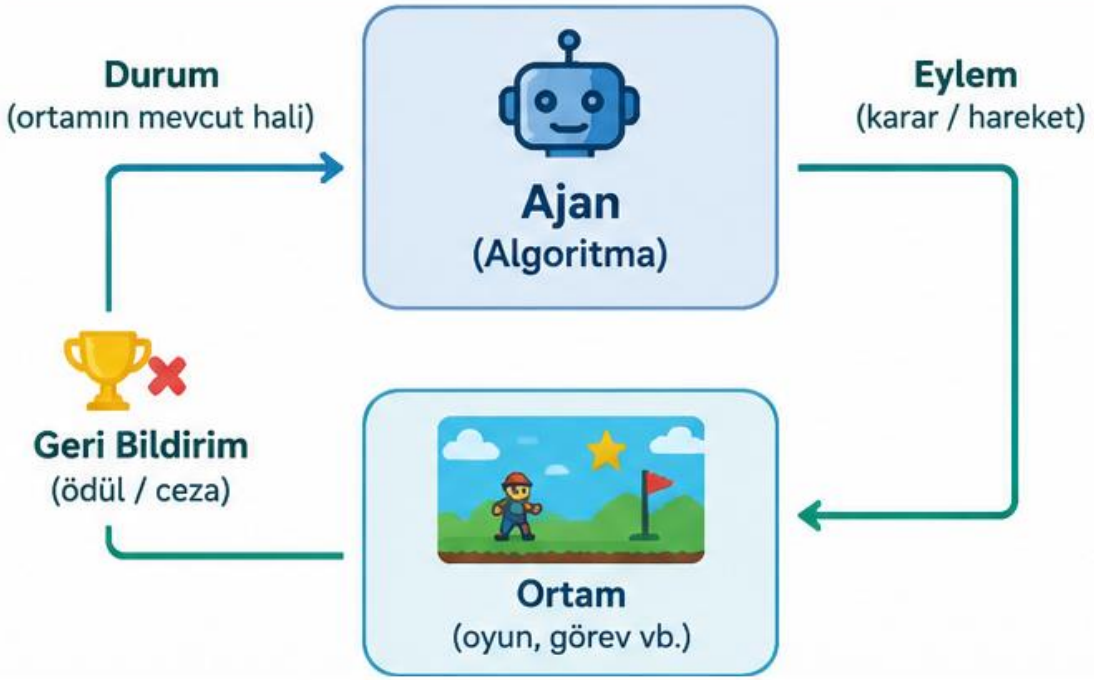
Yarı Denetimli Öğrenme

Örnek

Bir e-ticaret sitesinde, ürün görsellerinin bir kısmı etiketlenmiş (kategori bilgisi var), büyük bir kısmı ise etiketlenmemiştir. Yarı denetimli öğrenme, hem etiketli hem de etiketsiz verileri kullanarak ürünleri doğru kategorilere ayırmayı öğrenir.

Takviyeli Öğrenme

Algoritmanın eylem gerçekleştirerek ve geri bildirim alarak öğrendiği öğrenme türüdür. Amaç, bir ödülü en üst düzeye çıkarmaktır, örneğin bir oyunu kazanmak veya bir görevi tamamlamak.



Yapay Zeka ve Uygulamaları - Copyright © 2026

Takviyeli Öğrenme

Örnek

Bir robot, labirentteki hedefe ulaşmak için farklı eylemler dener. Hedefe ulaştığında ödül alır, yanlış yolda giderse ceza alır. Zamanla, aldığı geri bildirimlere göre en iyi stratejiyi öğrenir ve ödülü en üst düzeye çıkarmayı başarır.

Yarı Denetimli Öğrenme: Hem etiketli hem de etiketsiz verilerin kullanıldığı öğrenme türüdür.

Takviyeli Öğrenme: Algoritmanın eylem gerçekleştirerek ve geri bildirim alarak öğrendiği öğrenme türüdür. Amaç, bir ödülü en üst düzeye çıkarmaktır, örneğin bir oyunu kazanmak veya bir görevi tamamlamaktır.

Bu türler, makine öğrenimi algoritmalarının farklı uygulamaları için kullanılan temel yaklaşımları temsil eder.

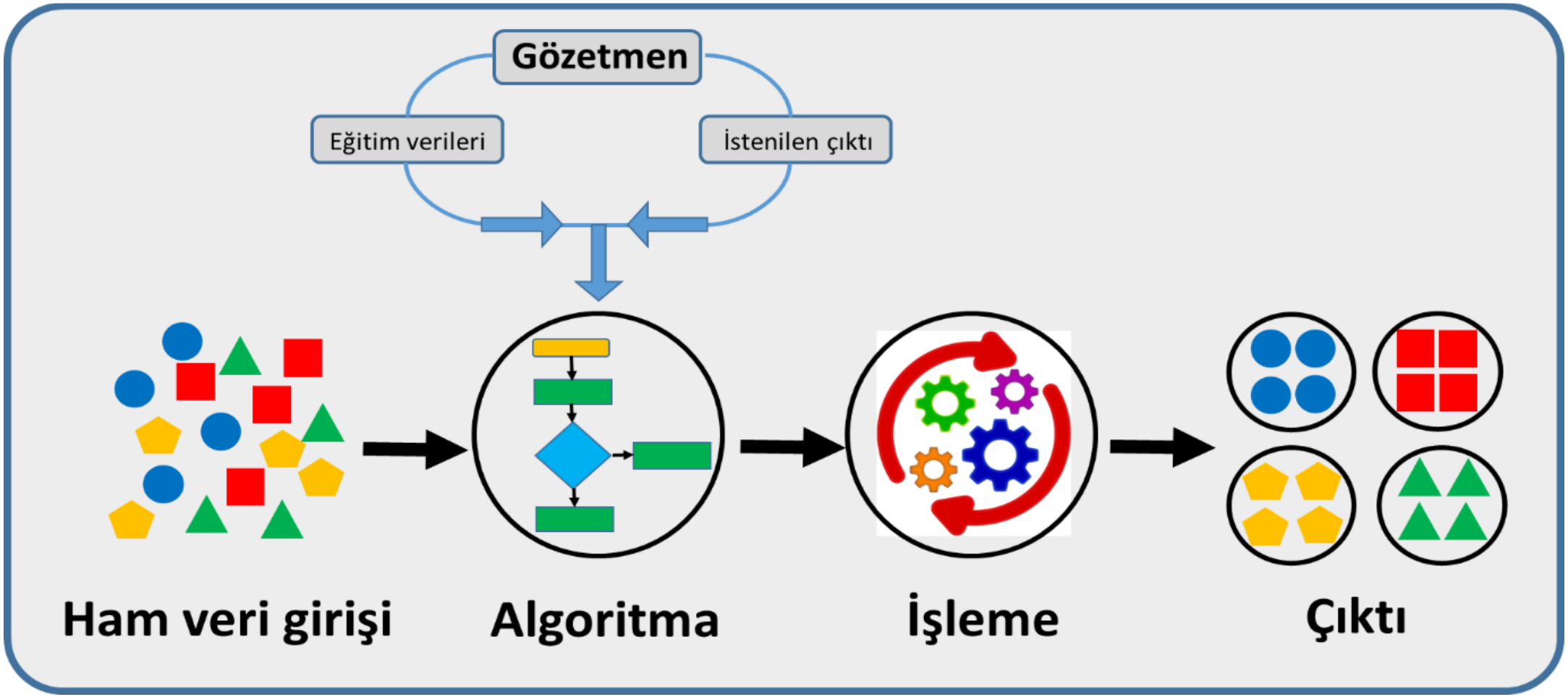
Yapay Zeka ve Uygulamaları - Copyright © 2026

7.2.1. Denetimli Makine Öğrenmesi

Denetimli makine öğrenmesi algoritmaları, doğru şekilde etiketlenmiş verilere ve bir geliştiricinin veya programcının gözetimine dayanır (Hutchison, 2021).

Etiketli veriler, makinenin eğitim sırasında girdi verilerini ve buna karşılık gelen çıktıyı bildiği, ardın-dan test sürecinde çıktıyı tahmin ettiği anlamına gelir. Düzgün eğitilmiş modeller, yeni bir girdi veri seti için gerçek dünya çıktılarına yakın, doğru bir tahmin sağlamalıdır (Salah, 2024).

Model, giriş verilerini istenen çıktıyla eşleştirir ve model yüksek bir doğruluk düzeyine ulaşana kadar eğitilir (Hutchison, 2021).



Şekil 7.5. Denetimli makine öğrenmesi

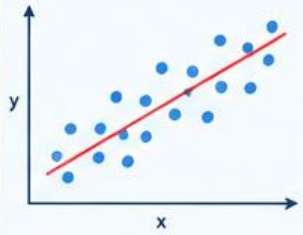
Denetimli öğrenmenin iki ana türü vardır: regresyon ve sınıflandırmadır. Regresyon, çıktı değişkeni gerçek bir değer olduğunda, örneğin büyüklüğü ve konumu göz önüne alındığında bir evin fiyatını tahmin etmek için kullanılır. Bir e-postanın spam olup olmadığının belirlenmesi gibi çıktı değişkeni bir kategori olduğunda sınıflandırma kullanılır (Pardeshi, 2023).

En popüler kullanılan denetimli öğrenme algoritmalarından bazıları şunlardır:

- Linear Regression
- Logistic Regression
- Support Vector Machine
- K Nearest Neighbor
- Decision Tree
- Random Forest
- Naive Bayes

1 Linear Regression

Sürekli (sayısal) hedef değişkenleri tahmin etmek için kullanılan en basit ve en yaygın algoritmalardan biridir. Veriler arasında lineer bir ilişki varsayar.



Doğrusal regresyon örneği

Öne Çıkan Özellikler

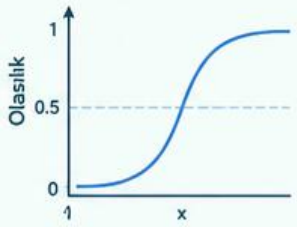
- Sürekli değer tahmini
- Kolay yorumlanabilir
- Hızlı ve düşük hesaplama maliyeti

Kullanım Alanları

Fiyat tahmini, satış analizi, trend analizi

2 Logistic Regression

İkili (binary) sınıflandırma problemlerinde kullanılır. Verilerin belirli bir sınıfa ait olma olasılığını tahmin eder. Genellikle sınıflandırma problemlerinde tercih edilir.



Lojistik regresyon eğrisi

Öne Çıkan Özellikler

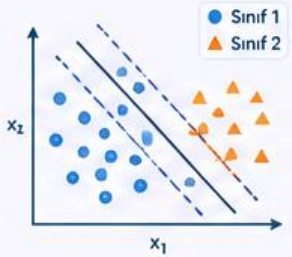
- İkili sınıflandırma
- Olasılık tahmini
- Yorumlanabilir sonuçlar

Kullanım Alanları

Dolandırıcılık tespiti, hasta teşhisi, spam filtreleme

3 Support Vector Machine

Sınıfları en iyi ayıran sınırları (hiper düzlemleri) bulmaya çalışır. Özellikle küçük ve orta ölçekli verilerde yüksek başarı gösterir.



SVM karar sınırı ve marjları

Öne Çıkan Özellikler

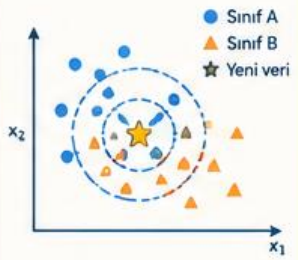
- Doğrusal ve doğrusal olmayan sınıflandırma
- Yüksek genelleme başarısı
- Gürültüye dayanıklı

Kullanım Alanları

Yüz tanıma, metin sınıflandırma, öneri sistemleri

4 K Nearest Neighbor

Yeni bir verinin sınıfı, veri setindeki en yakın komşularının sınıflarına göre belirlenir. Basit ama etkili bir sınıflandırma algoritmasıdır.



KNN komşuluk ilişkisi

Öne Çıkan Özellikler

- Basit ve anlaşılır
- Karmaşık veri yapılarıyla uyumlu
- Parametre ayarı (k) önemli

Kullanım Alanları

Patern tanıma, müşteri segmentasyonu, tıbbi görüntü analizi

5 Decision Tree

Verileri, belirli özelliklere göre dallara ayırarak sınıflandırma veya regresyon yapan ağaç yapılı bir algoritmadır. Karar kuralları kolayca yorumlanabilir.



Karar ağacı örneği

Öne Çıkan Özellikler

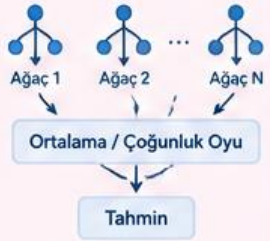
- Kolay yorumlanabilir
- Hem sınıflandırma hem regresyon
- Eksik verilere dayanıklı

Kullanım Alanları

Kredi skorlaması, risk analizi, medikal karar destek

6 Random Forest

Birden fazla karar ağacının birlikte çalıştığı topluluk (ensemble) yöntemidir. Daha yüksek doğruluk ve daha iyi genelleme performansı sağlar.



Random forest yapısı

Öne Çıkan Özellikler

- Yüksek doğruluk
- Aşırı öğrenmeyi azaltır
- Özellik önemini ölçebilir

Kullanım Alanları

Orman yangını tahmini, hisse senedi analizi, sınıflandırma problemleri

7 Naive Bayes

Bayes teoremine dayanır ve özelliklerin birbirinden bağımsız olduğunu varsayar. Hızlıdır ve özellikle metin sınıflandırma gibi problemlerde yaygın olarak kullanılır.

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)}$$



Naive Bayes modeli

Öne Çıkan Özellikler

- Hızlı ve verimli
- Az veriyle iyi sonuç verir
- Metin sınıflandırmada çok başarılı

Kullanım Alanları

Spam filtreleme, duygu analizi, belge sınıflandırma

Yapay Zeka ve Uygulamaları - Copyright © 2026

Denetimli öğrenme, makine öğreniminin en popüler ve tipik örneği olarak kabul edilir. Anlaşılması kolay, makine öğreniminin en kolay ve en basit şeklidir. Bu, bir çocuğa flaş kartların yardımıyla nesneleri tanımayı öğretmek gibidir (PathakJan, 2021).

Denetimli makine öğrenimi, görüntü sınıflandırma, konuşma tanıma, doğal dil işleme ve daha pek çok uygulamada yaygın olarak kullanılmaktadır (Pardeshi, 2023).

7.2.1.1. Sınıflandırma

Bir diğer makine öğrenimi yaklaşımı da sınıflandırmadır. Bu, giriş verilerini kategorilere ayıran makine öğrenimi tekniklerinden biridir. Her giriş değişkeni, bir kategoriyi işaretleyen bir etiket alır. Başka bir deyişle, sınıflandırma tekniği, giriş verilerini kategorik çıkış etiketlerinden birine eşlemek için kullanılır. Bu tür makine öğrenimi tekniği genellikle sahtekârlık tespiti ve pazarlama, hastalık teşhisi ve diğer alanlarda kullanılır (Danielkievich, 2023).

Sınıflandırma, belirli bir veri kümesini önceden tanımlanmış sınıflara veya gruplara ayırma işlemidir. Denetimli bir öğrenme tekniğidir; bu, modelin etiketli veriler üzerinde eğitildiği ve amacın yeni, görünmeyen verilerden oluşan sınıfı veya grubu tahmin et-mek olduğu anlamına gelir (Pardeshi, 2023).

Sınıflandırma (Classification)

Sınıflandırma, giriş verilerini belirli kategorilere ayıran makine öğrenimi tekniklerinden biridir. Her giriş değişkeni, bir kategoriyi işaretleyen bir etiket alır. Kısacası, sınıflandırma tekniği, giriş verilerini kategorik çıkış etiketlerinden birine eşlemek için kullanılır.

Sınıflandırma Kullanım Alanları



Sahtekârlık Tespiti

Kredi kartı işlemlerinde şüpheli durumları belirleme.



Pazarlama

Müşteri segmentasyonu ve hedef kitle analizi yapma.



Hastalık Teşhisi

Tıbbi verilerle hastalık sınıflandırması yapma.



Diğer Alanlar

Metin sınıflandırma, spam tespiti, duygu analizi gibi birçok alanda kullanılır.



Özet

- Sınıflandırma, giriş verilerini önceden tanımlanmış sınıflara ayırır.
- Denetimli bir öğrenme tekniğidir.
- Model, etiketli verilerle eğitilir ve yeni verilerde sınıf tahmini yapar.
- Sahtekârlık tespiti, pazarlama, hastalık teşhisi gibi birçok alanda yaygın olarak kullanılır.

(Danielki-evich, 2023) (Pardeshi, 2023)

Nasıl Çalışır?

1. **Etiketli verilerle eğitim**
   Model, her giriş verisi için doğru sınıf etiketini içeren etiketli veriler üzerinde eğitilir.

2. **Öğrenme süreci**
   Model, verilerdeki örüntüleri ve ilişkileri öğrenir.

3. **Yeni verilerde tahmin**
   Eğitimden sonra, model yeni ve görünmeyen veriler için sınıf tahmini yapar.

Eğitim Verisi (Etiketli)

Yaş	Gelir	Kredi Notu	Sınıf
25	50.000	700	İyi
40	30.000	600	Riskli
32	80.000	750	İyi
28	20.000	500	Riskli
...	...	...	...

Sınıf Etiketleri

- İyi
- Riskli
- Dolandırıcılık

Yeni Veri (Etiketsiz)

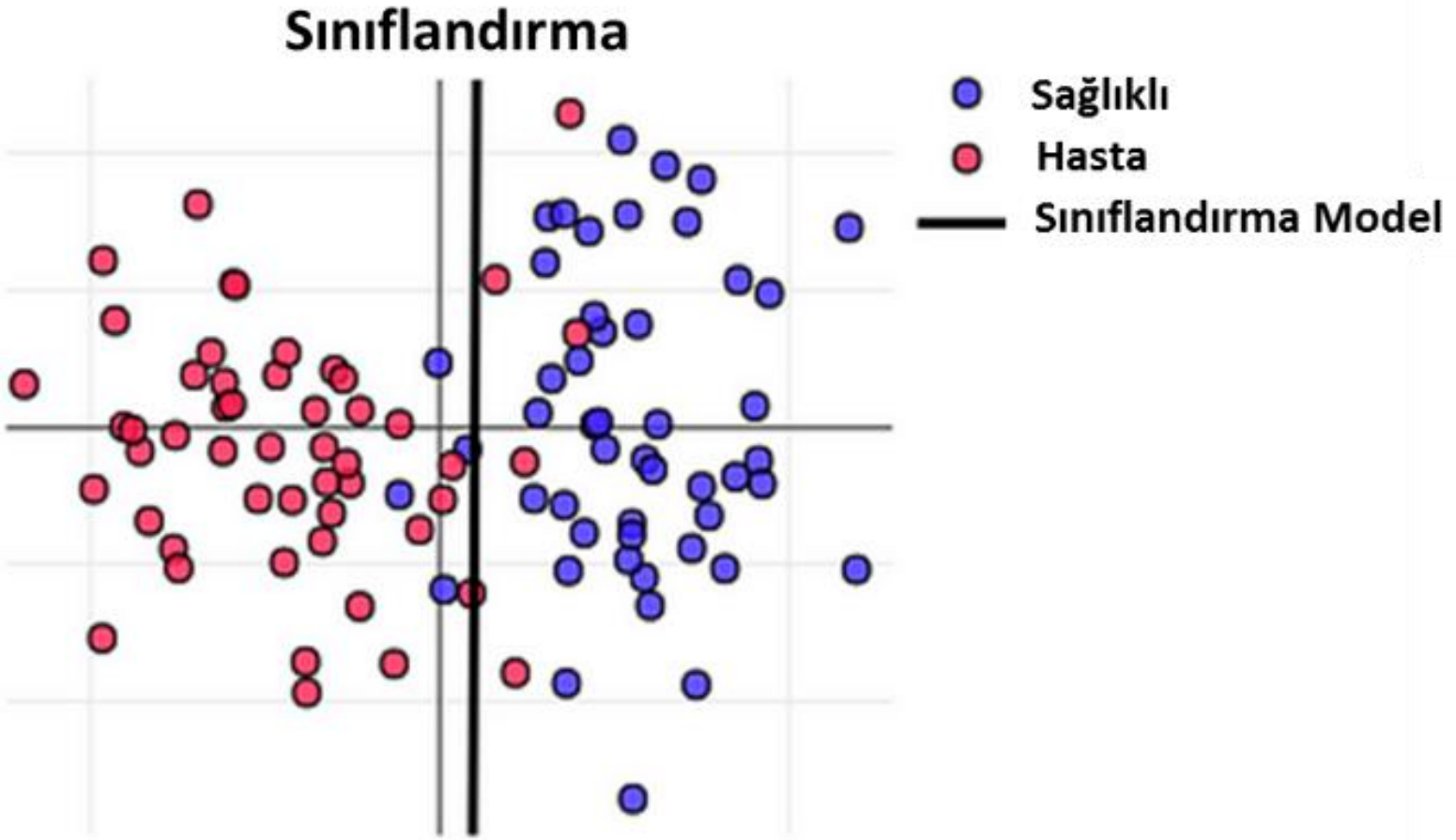
Yaş	Gelir	Kredi Notu
35	60.000	680



Tahmin Sonucu

İyi

Yapay Zeka ve Uygulamaları - Copyright © 2026



Şekil 7.6. Sınıflandırma

İki ana sınıflandırma problemi türü vardır: ikili sınıflandırma ve çok sınıflı sınıflandırma. İkili sınıflandırmada veriler "pozitif" ve "negatif" olmak üzere iki sınıfa ayrılır. Çok sınıflı sınıflandırmada veriler "kırmızı", "yeşil" ve "mavi" gibi ikiden fazla sınıfa ayrılır (Pardeshi, 2023).

Bilinen bazı sınıflandırma algoritmaları arasında Rastgele Orman Algoritması, Karar Ağacı Algoritması, Lojistik Regresyon Algoritması ve Destek Vektör Makinesi Algoritması bulunur (Kanade, 2022).

Sınıflandırma algoritmaları, verilerde temsil edildiği şekliyle bir varlığın, nesnenin veya olayın kategorisine karar verir. En basit sınıflandırma algoritmaları evet/hayır, satış/satış değil veya kedi/kedi değil gibi ikili soruları yanıtlar. Daha karmaşık algoritmalar, nesneleri kedi, köpek veya fare gibi birden fazla kategoriye ayırır (Lawton, 2023).

Yapay Zeka ve Uygulamaları - Copyright © 2026

Verileri daha fazla kategorize etmek için kullanılan sınıflandırma algoritmaları, bu verileri sıralamak ve hatta gizlemek için harika bir araçtır (Biba, 2023).

Sınıflandırma algoritmaları, görüntü ve konuşma tanıma, doğal dil işleme ve biyoenformatik gibi çeşitli uygulamalarda yaygın olarak kullanılmaktadır (Pardeshi, 2023).

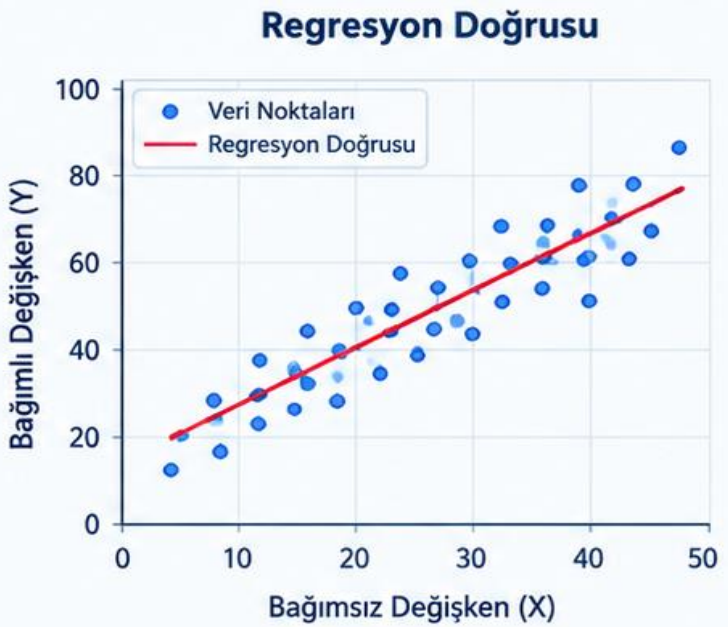
7.2.1.2. Regresyon

Regresyon analizi, değişkenler arasındaki ilişkiyi anlamak ve bu ilişkiyi kullanarak bir değişkenin diğerlerine nasıl bağlı olduğunu tahmin etmek için kullanılan istatistiksel bir tekniktir. Genellikle, bağımlı bir değişkenin bir veya daha fazla bağımsız değişken tarafından nasıl etkilendiğini anlamak için kullanılır.

Regresyon, çıktı değerini yeniden üretmeyi amaçlayan bir tekniktir. Örneğin, belirli bir şehirdeki bir evin fiyatı veya bir hisse senedinin değeri gibi bazı ürünlerin fiyatını tahmin etmek için bunu kullanabiliriz. İstersek tahmin edebileceğimiz çok sayıda şey vardır (Mrukwa, 2023). Regresyon analizi, geçmiş verilere dayalı olarak sayısal bir değerin tahmin edilmesiyle ilgilidir (Arya, 2024). Regresyonun amacı, girdi değişkenleri ile hedef değişken arasındaki ilişkiyi tanımlayan en uygun matematiksel modeli bulmaktır (Pardeshi, 2023).

Regresyon

Regresyon analizi, değişkenler arasındaki ilişkiyi anlamak ve bu ilişkiyi kullanarak bir değişkenin diğerlerine nasıl bağlı olduğunu tahmin etmek için kullanılan istatistiksel bir tekniktir. Genellikle, bağımlı bir değişkenin bir veya daha fazla bağımsız değişken tarafından nasıl etkilendiğini anlamak için kullanılır.



Örnek Regresyon Denklemi

$$y = \beta_0 + \beta_1 x + \varepsilon$$

- y : Bağımlı değişken (tahmin edilen değer)
- x : Bağımsız değişken (girdi değişkeni)
- β_0 : Sabit terim
- β_1 : Regresyon katsayısı
- ε : Hata terimi

Amaç

Girdi değişkenleri ile hedef değişken arasındaki ilişkiyi tanımlayan en uygun matematiksel modeli bulmaktır.

Ne İşe Yarar?

Çıktı değerini yeniden üretmeyi amaçlar ve gelecekteki değerleri tahmin etmek için bullanılır.

Örnek Kullanım Alanları

Belirli bir şehirdeki bir evin fiyatı, bir hisse senedinin değeri gibi tahminler yapılabilir. İstediğimiz birçok farklı ürünü ve olayı tahmin edebiliriz (Mrukwa, 2023).

Temel Özellik

Regresyon analizi, geçmiş verilere dayalı olarak sayısal bir değerin tahmin edilmesiyle ilgilenir (Arya, 2024).

Regresyonun Kullanım Örnekleri



Ev Fiyatı Tahmini

(Bağımsız değişken: metrekare, bağımlı değişken: fiyat)



Hisse Senedi Değeri Tahmini

(Bağımsız değişken: geçmiş fiyatlar, bağımlı değişken: gelecek fiyat)



Satış Tahmini

(Bağımsız değişken: reklam harcaması, bağımlı değişken: satış miktarı)

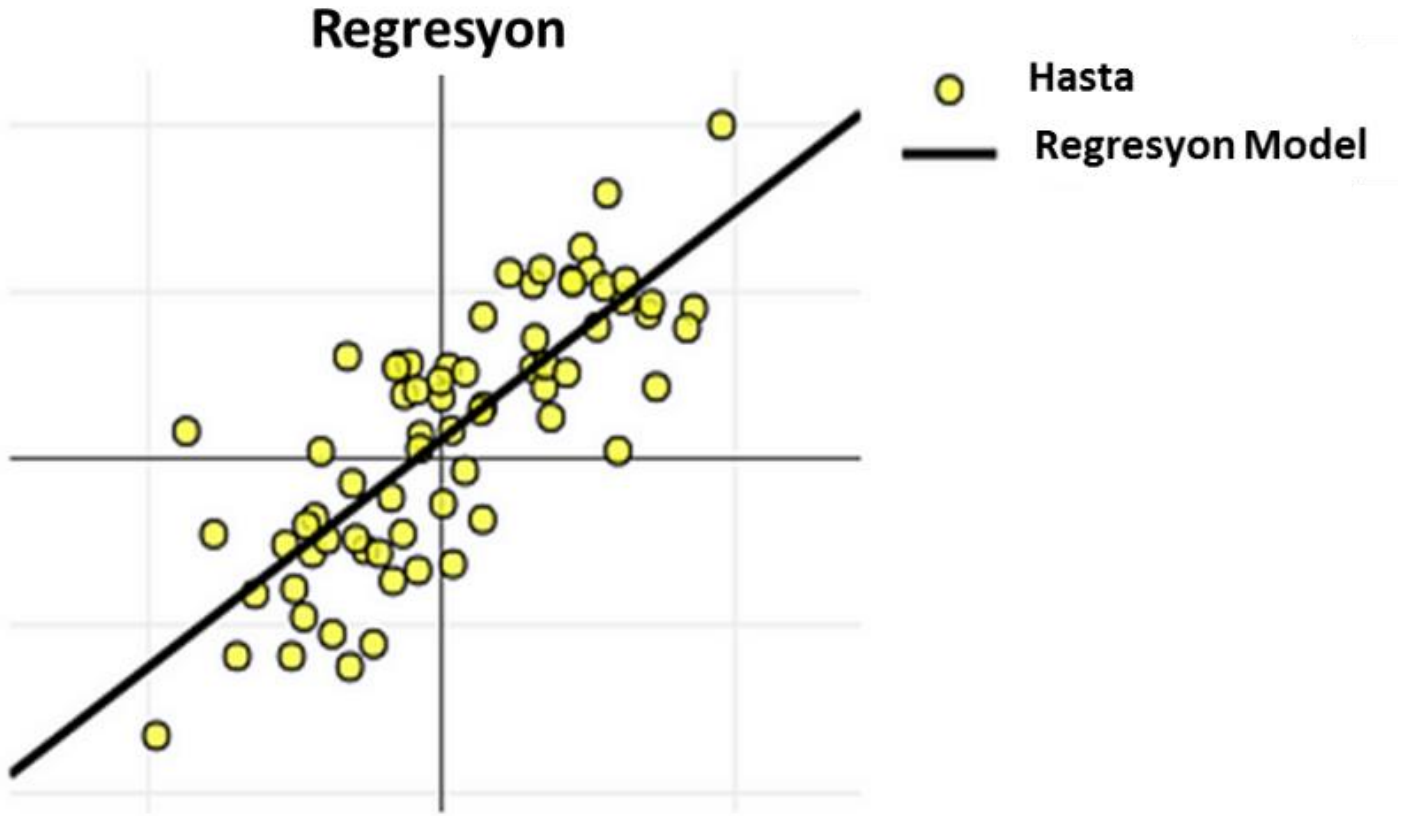


Sağlık Verisi Analizi

(Bağımsız değişken: yaş, kilo, bağımlı değişken: tansiyon)



Regresyon analizi, verilerdeki ilişkiyi ortaya koyarak geleceğe yönelik tahminlerde bulunmamıza olanak sağlar.



Şekil 7.7. Regresyon model

Çok sayıda regresyon analizi tekniği türü vardır ve her yöntemin kullanımı faktörlerin sayısına bağlıdır. Bu faktörler arasında hedef değişkenin türü, regresyon çizgisinin şekli ve bağımsız değişkenlerin sayısı yer alır.

Farklı regresyon teknikleri:

- Doğrusal Regresyon (Linear Regression)
- Lojistik regresyon (Logistic Regression)
- Sırt Regresyon (Ridge Regression)
- Kement Regresyon (Lasso Regression)
- Polinom Regresyon (Polynomial Regression)
- Bayesian Doğrusal Regresyon (Bayesian Linear Regression)

Regresyon Analizi Teknikleri

Farklı veri yapıları ve problem türleri için farklı regresyon teknikleri kullanılır.

Çok sayıda regresyon analizi tekniği türü vardır ve her yöntemin kullanımı faktörlerin sayısına bağlıdır. Bu faktörler arasında hedef değişkenin türü, regresyon çizgisinin şekli ve bağımsız değişkenlerin sayısı yer alır.

Hangi faktörler belirleyicidir?

- Hedef değişkenin türü (sürekli, kategorik vb.)
- Regresyon çizgisinin şekli (doğrusal, eğrisel vb.)
- Bağımsız değişkenlerin sayısı (tek veya çok değişkenli)

Yapay Zeka ve Uygulamaları - Copyright © 2026

Hangi Yöntemi Ne Zaman Kullanmalı?

Doğrusal Regresyon
→ Sürekli hedef değişken, basit model

Lojistik Regresyon
→ Kategorik hedef değişken, sınıflandırma

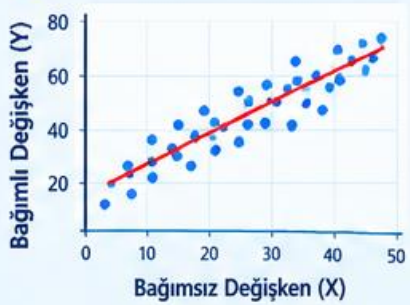
Ridge Regresyon
→ Çoklu doğrusal bağlantı, yüksek değişken sayısı

Lasso Regresyon
→ Değişken seçimi, model sadeleştirme

Polinom Regresyon
→ Doğrusal olmayan ilişkiler, karmaşık veri

Bayesian Regresyon
→ Belirsizlik analizi, küçük veri setleri

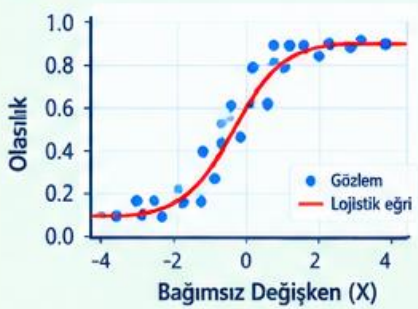
1 Doğrusal Regresyon (Linear Regression)



- Sürekli bir hedef değişken için kullanılır.
- İki veya daha fazla değişken arasındaki doğrusal ilişkiyi modellemek için idealdir.
- En yaygın kullanılan regresyon türüdür.

$$y = \beta_0 + \beta_1 x + \varepsilon$$

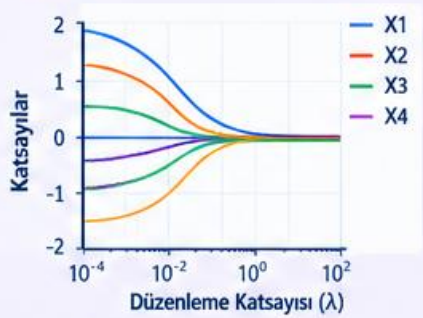
2 Lojistik Regresyon (Logistic Regression)



- Kategorik (ikili veya çok sınıflı) hedef değişkenler için kullanılır.
- Olasılık tahmini yapar.
- Sınıflandırma problemlerinde yaygındır.

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

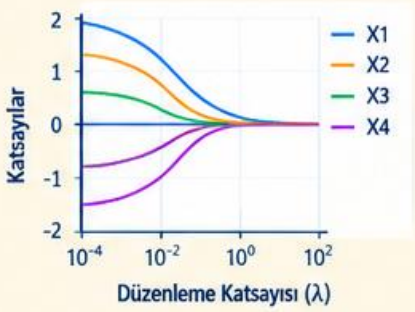
3 Sırt Regresyon (Ridge Regression)



- Çoklu doğrusal bağlantı sorununu azaltır.
- Katsayıları küçülterek modelin genelleme yeteneğini artırır.
- Özellikle çok sayıda bağımsız değişken olduğunda tercih edilir.

$$min \ \sum (y - X\beta)^2 + \lambda \sum \beta^2$$

4 Kement Regresyon (Lasso Regression)



- Değişken seçimi yapar (bazı katsayıları sıfıra indirir).
- Modeli daha basit hale getirir.
- Yüksek boyutlu veri setlerinde etkilidir.

$$min \ \sum (y - X\beta)^2 + \lambda \sum |\beta|$$

5 Polinom Regresyon (Polynomial Regression)



- Doğrusal olmayan ilişkileri modellemek için kullanılır.
- Bağımsız değişkenin polinom terimleri eklenir.
- Karmaşık veri yapılarında daha iyi uyum sağlar.

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_n x^n$$

6 Bayesian Doğrusal Regresyon (Bayesian Linear Regression)



- Önsel bilgi (prior) ile tahmin yapar.
- Belirsizliği (güven aralığı) dikkate alır.
- Küçük veri setlerinde daha etkili olabilir.

$$p(\beta \mid D) \propto p(D \mid \beta) \cdot p(\beta)$$

Yapay Zeka ve Uygulamaları - Copyright © 2026

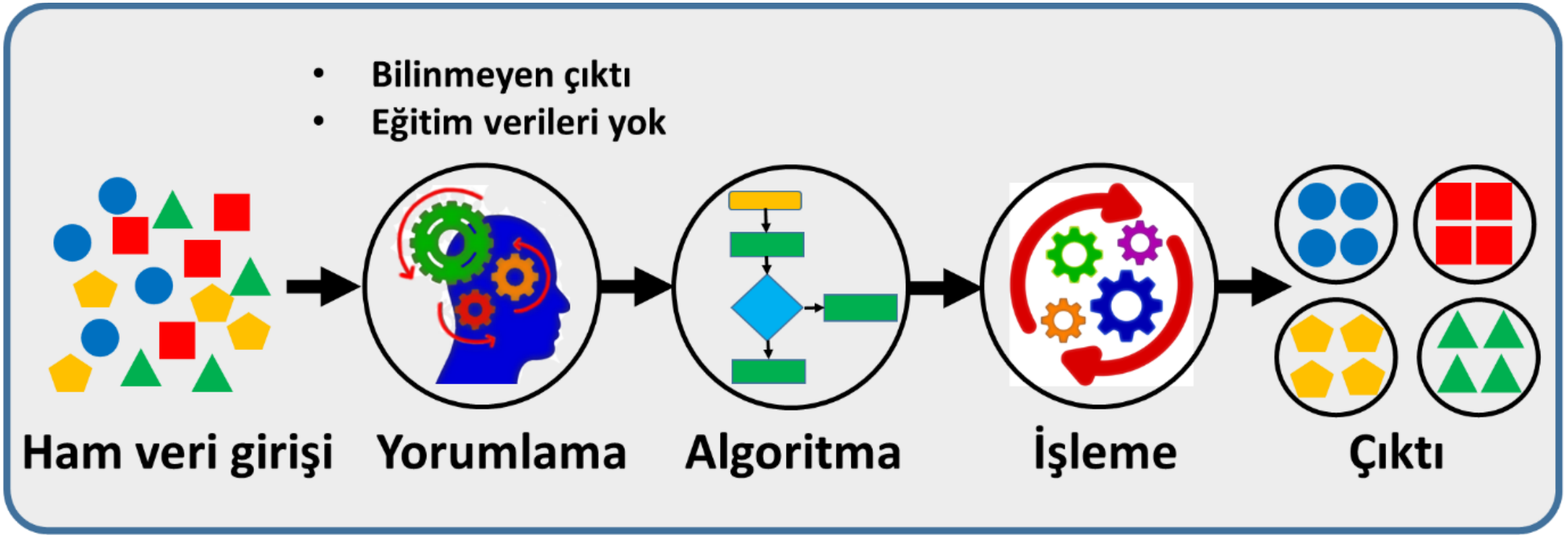
Regresyon modelleri finans, ekonomi ve mühendislik dahil birçok alanda yaygın olarak kullanılmaktadır. Ayrıca borsa tahmini, hava durumu tahmini ve tıbbi teşhis gibi uygu-lamalarda da kullanılırlar. Regresyon modelleri, farklı değişkenler arasındaki ilişkiyi anlamak ve gelecekteki sonuçlara ilişkin tahminler yapmak için kullanışlıdır (Pardeshi, 2023).

7.2.2. Denetimsiz Makine Öğrenmesi

Denetimli öğrenmenin aksine, denetimsiz makine öğrenimi eğitim sırasında denetim gerektirmez. Denetimsiz algoritmalarda, makine etiketlenmemiş verilere maruz kalır ve bu verilerin ne olduğunu kendi kendine öğretmesi beklenir (Salah, 2024).

Denetimsiz öğrenme, etiketli verilerin mevcut olmadığı veya elde edilmesinin çok pahalı olduğu durumlarda faydalıdır. Denetimli öğrenmeyi kullanmadan önce bir ön işleme adımı olarak da kullanılabilir (Pardeshi, 2023).

Bu makine öğrenimi yöntemi, denetimli öğrenmenin tam tersidir ve hiçbir insan katılımı gerektirmez. Denetimsiz makine öğrenimi algoritmaları altında, makine öğrenimi modelini eğitmek için etiketlenmemiş verileri kullanırsınız. ML algoritması verilerden öğrenir ve kalıpları, eğilimleri veya ilişkileri tanımlar. Girdiyi çıktı katmanına eşlemez; bu modeli öğretmek için yalnızca girdi verileri gereklidir (Danielkievich, 2023).



Şekil 7.8. Denetimsiz makine öğrenmesi

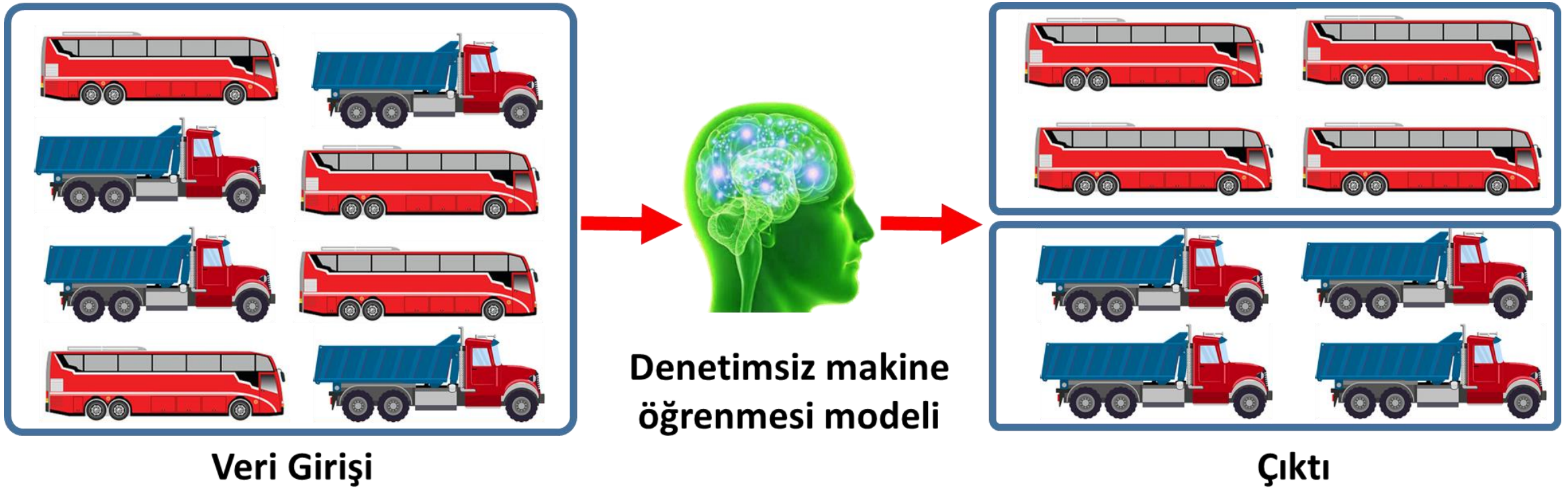
Yapay Zeka ve Uygulamaları - Copyright © 2026

Denetimsiz öğrenme, modelin etiketli eğitim verileriyle sağlanmadığı bir makine öğrenimi türüdür. Bunun yerine modele bir veri kümesi verilir ve veriler içindeki kalıpları veya yapıları kendi başına bulması gerekir. Denetimsiz öğrenmenin en yaygın türleri kümeleme (clustering), birliktelik (assocation) ve boyut indirgemedir (Pardeshi, 2023).

Kümeleme algoritmaları benzer veri noktalarını bir arada gruplandırırken boyut indirgeme algoritmaları, önemli ilişkileri koruyan verilerin daha düşük boyutlu bir temsilini bulmaya çalışır (Pardeshi, 2023). Birliktelik (assocation), makinelerin büyük girdi veri kümelerindeki değişkenler arasındaki ilginç ilişkileri ve bağlantıları keşfettiği denetimsiz bir öğrenme tekniğidir. İşletmeler, web kullanım madenciliği, intihal kontrolü, öneriler geliştirme ve sürekli üretim gibi uygulamalar için çağrışımları yaygın olarak kullanır (Joshi, 2022).

Denetimsiz öğrenme algoritmalarının bazı örnekleri K-Ortalamalar, Hiyerarşik kümeleme, PCA (Temel Bileşen Analizi) ve Otomatik Kodlayıcılardır (Pardeshi, 2023).

Denetimsiz öğrenme, beklenen çıktıya ilişkin hiçbir örnek veya öneri içermeyen girdi verileriyle çalışan bir algoritma türüdür (Anisimova, 2023).



Şekil 7.9. Denetimsiz makine öğrenmesi örneği

Yapay Zeka ve Uygulamaları - Copyright © 2026

Şekil 8.'de görüldüğü gibi, otobüs mü yoksa kamyon mu olduğunu sınıflandırmak için araç görüntülerini kullanan bir denetimsiz öğrenme tekniği örneği gösterilmektedir. Model, aracın uzunluğu ve genişliği, ön ve arka uç kapakları, tavan kaputları, kullanılan tekerlek türleri vb. gibi aracın parçalarını tanımlayarak öğrenir. Bu özelliklere dayanarak model, aracın otobüs mü yoksa kamyon mu olduğunu sınıflandırır (Menon, 2023).

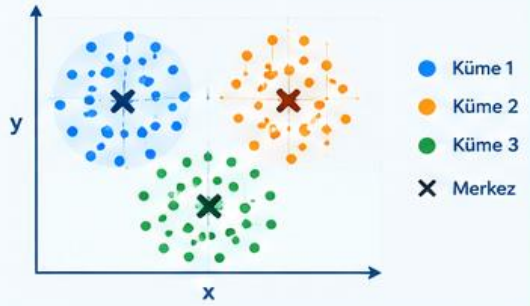
Denetimsiz öğrenmenin yaygın örneklerinden bazıları şunlardır:

- K Means Clusterin
- Hierarchical Clustering
- DBSCAN
- Principal Component Analysis

Yapay Zeka ve Uygulamaları - Copyright © 2026

1 K Means Clustering

Verileri, benzer özelliklere sahip k kümeye ayıran basit ve etkili bir kümeleme algoritmasidır.



Nasıl Çalışır?

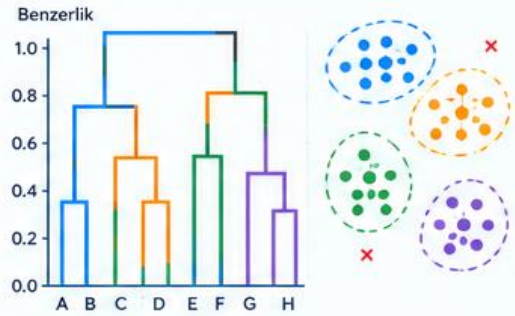
1. K (küme sayısı) belirlenir.
2. Rastgele merkezler seçilir.
3. Veriler en yakın merkeze atanır.
4. Merkezler güncellenir.
5. Yakınsama sağlanana kadar adımlar tekrarlanır.

Kullanım Alanları

- Müşteri segmentasyonu
- Görüntü işleme
- Doküman kümeleme

2 Hierarchical Clustering

Verileri, benzerliklerine göre hiyerarşik bir yapı oluşturarak gruplandıran kümeleme yöntemidir.



Nasıl Çalışır?

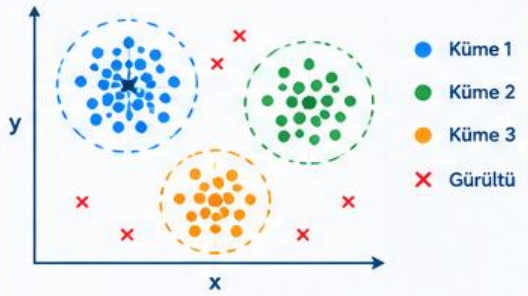
1. Her veri noktası kendi kümesini oluşturur.
2. En yakın kümeler birleştirilir.
3. Bu işlem, tek bir küme kalana kadar tekrarlanır.
4. Sonuç, bir dendrogram (ağaç diyagramı) ile görselleştirilir.

Kullanım Alanları

- Veri analizi ve keşif
- Biyoinformatik
- Müşteri analizi

3 DBSCAN

Yoğunluk tabanlı bir kümeleme algoritmasıdır. Veri noktalarının yoğun bölgelerini küme olarak belirler ve izole noktaları (gürültü) tespit eder.



Nasıl Çalışır?

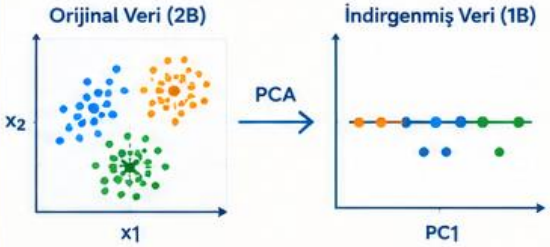
1. Epsilon (ε) ve minimmum komşu sayısı (MinPts) belirlenir.
2. Yoğun bölgeler küme olarak tanımlanır.
3. Yoğunluk bağlantısı olan noktalar aynı kümeye atanır.
4. Yoğunluk eşiğini sağlamayan noktalar gürültü olarak işaretlenir.

Kullanım Alanları

- Anomali tespiti
- Coğrafi veri analizi
- Büyük ölçekli veri setleri

4 Principal Component Analysis

Yüksek boyutlu verileri, en fazla varyansı açıklayan daha az sayıda yeni boyuta indirgemeyi sağlayan bir boyut azaltma yöntemidir.



Nasıl Çalışır?

1. Veri setinin kovaryans matrisi hesaplanır.
2. Özdeğerler ve özvektörler bulunur.
3. En büyük varyansı açıklayan bileşenler seçilir.
4. Veriler, bu yeni bileşenlere projekte edilir.

Kullanım Alanları

- Görüntü ve ses işleme
- Veri görselleştirme
- Boyut indirgeme

Yapay Zeka ve Uygulamaları - Copyright © 2026

Denetimsiz öğrenme, makine öğrenme tekniklerinin dört ana türünden biridir. İnsan müdahalesi olmadan tahminlerde bulunmak için etiketlenmemiş ve sınıflandırılmamış veri kümelerini kullanır. Bu yöntem, sıralanmamış verileri benzerliklere ve farklılıklara göre kategorize etmek veya gruplamak, ayrıca girdi verilerindeki gizli kalıpları ve eğilimleri keşfetmek için kullanışlıdır (Joshi, 2022).

Denetimli öğrenmenin aksine, denetimsiz öğrenme algoritmaları herhangi bir sabit çıktı değişkeni olmadan verilerden öğrenir ve bu da onları karmaşık görevlere uygun hale getirir. Verilerdeki gizli kalıpları veya gruplamaları belirlerler; bu onları kümeleme, anormallik tespiti ve keşifsel veri analizi için ideal kılar (Joshi, 2022).

Denetimsiz Öğrenme Uygulamaları

- Dolandırıcılık işlemlerini tanımlamak için anormallik tespiti
- Tahmin ve tahmine dayalı modelleme
- Pazar bölümlerini belirleme ve hedefleme
- Ürün veya hizmet önermek için öneri sistemleri

Denetimsiz bir öğrenme algoritması, sıralanmamış veri kümesini girdinin benzerlikleri-ne, farklılıklarına ve modellerine göre gruplandırmayı amaçlar (Kanade, 2022).

7.2.2.1. Kümeleme (Clustering)

Kümeleme algoritmaları, denetimsiz makine öğreniminin en yaygın kullanılan örneğidir (Biba, 2023). Kümeleme benzerlikleri ve farklılıkları bulmak için kullanılır. Benzer şeyleri bir arada gruplandırır (Mrukwa, 2023). Örneğin müşterileri satın aldıkları ürünlere göre gruplandırmaktır (Kanade, 2022).

Kümeleme, makine öğrenimi mühendislerinin etiketlenmemiş verilerle çalışırken kalıpları ve yapıları tespit etmesine yardımcı olur. Bu makine öğrenimi yöntemi biyoloji, görüntü işleme, pazarlama ve diğer alanlarda kullanılır. E-ticaret geliştiricileri bu yaklaşımı müşteri ilgi alanlarına ve davranışlarına göre segmentasyona uygular. Özellik çıkarma işlemini başlattığımızda veya verileri sıkıştırdığımızda da kümeleme yöntemlerini uygulayabiliriz (Danielkievich, 2023).

Kümeleme (Clustering)

Kümeleme algoritmaları, denetimsiz makine öğreniminin en yaygın kullanılan örneğidir. Kümeleme, benzerlikleri ve farklılıkları bulmak için kullanılır. Benzer şeyleri bir arada gruplandırır. Etiketlenmemiş verilerdeki kalıpları ve bölgeleri tespit etmeye yardımcı olur.

Nasıl Çalışır?

1. **Veri Toplama**
   Etiketlenmemiş veriler toplanır.
2. **Özelliklerin Belirlenmesi**
   Verilerdeki önemli özellikler seçilir veya çıkarılır.
3. **Benzerlik Hesabı**
   Veriler arasındaki benzerlikler hesaplanır (mesafe ölçütleri).
4. **Küme Oluşturma**
   Algoritma, benzer verileri aynı kümeye yerleştirir.

Örnek: Müşteri Segmentasyonu



Kümeleme Nerelerde Kullanılır?

- **E-Ticaret**
  Müşteri segmentasyonu, ürün önerileri.
- **Biyoloji**
  Genetik veri analizi, hücre grupları.
- **Görüntü İşleme**
  Görüntü sınıflandırma, nesne tespiti.
- **Pazarlama**
  Müşteri grupları, pazar analizi.
- **Diğer Alanlar**
  Veri sıkıştırma, özellik çıkarma, segmentasyon.

Neden Önemlidir?

- Karmaşık veri yapılarındaki kalıpları ortaya çıkarır.
- Etiketlenmemiş verilerle çalışmaya olanak tanır.
- Müşteri ve kullanıcı davranışlarını anlamaya yardımcı olur.
- Karar verme süreçlerini destekler.
- Veri analizi, görselleştirme ve modelleme süreçlerinde önemli bir rol oynar.

Örnek Kullanım Senaryosu

Bir e-ticaret firması, müşterilerini satın aldıkları ürünlere göre segmentlere ayırarak her gruba özel kampanyalar sunabilir.

Yapay Zeka ve Uygulamaları - Copyright © 2026

1 Kümeleme Nedir?

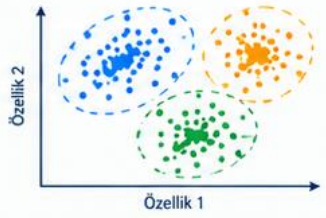
Kümeleme algoritmaları, denetimsiz makine öğreniminin en yaygın kullanılan örneğidir (Biba, 2023). Kümeleme, benzerlikleri ve farklılıkları bulmak için kullanılır. Benzer şeyleri bir arada gruplandırır (Mrukwa, 2023). Örneğin müşterileri satın aldıkları ürünlere göre gruplandırmak (Kanade, 2022).

Temel Amaç

- Etiketlenmemiş verilerdeki kalıpları ve yapıları tespit etmek.
- Verileri anlamlı gruplara ayırmak.
- Veri keşfi ve görselleştirme sağlamak.

2 Ne Zaman Kullanılır?

- Etiketli veri olmadığında
- Müşteri segmentasyonu (ör. e-ticaret)
- Görüntü işleme ve nesne gruplama
- Biyoloji (genetik, sınıflandırma)
- Pazarlama (hedef kitle analizi)
- Özellik çıkarma ve veri sıkıştırma
- Veri keşfi ve görselleştirme



Kümeleme, makine öğrenimi mühendislerinin etiketlenmemiş verilerle çalışırken kalıpları ve yapıları tespit etmesine yardımcı olur.

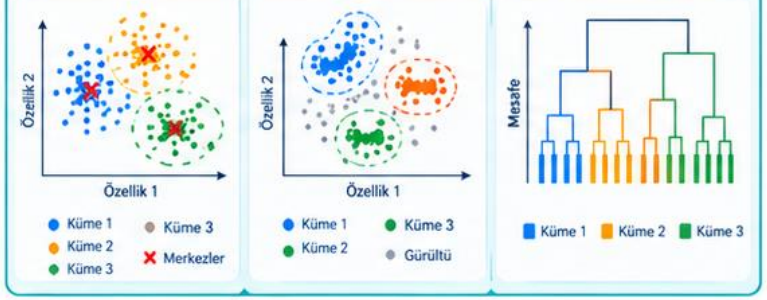
3 Başlıca Kümeleme Algoritmaları

1. **K-Means** – En yaygın kullanılan yöntemdir. Veri, K adet kümeye ayrılır.
2. **DBSCAN** – Yoğunluk temelli, farklı şekillerdeki kümeleri bulabilir.
3. **Hiyerarşik Kümeleme** – Ağaç yapısı (dendrogram) oluşturur.
4. **Gaussian Mixture Models (GMM)** – Olasılıksal kümeleme.
5. **Fuzzy C-Means** – Her veri noktası birden fazla kümeye ait olabilir.
6. **Spectral Clustering** – Grafik temelli kümeleme.
7. **Mean Shift** – Yoğunluk tepe noktalarını bulur.
8. **Agglomerative Clustering** – Alt kümeden üst kümeye birleştirir.
9. **BIRCH** – Büyük veri setleri için verimli bir yöntem.
10. **OPTICS** – Uzaklık tabanlı, yoğunluk temelli.

4 Yaygın Kümeleme Algoritmaları ve Özellikleri

Algoritma	Temel Özellik	Avantajları	Dezavantajları
K-Means	Merkez tabanlı	Hızlı, ölçeklenebilir	Küme sayısı önceden gerekir, şekil varsayımı (küresel)
DBSCAN	Yoğunluk tabanlı	Farklı şekiller, gürültüye dayanıklı	Parametre seçimi hassas
Hiyerarşik	Ağaç yapısı	Küme sayısı esnek, görselleştirme kolay	Büyük veri setlerinde yavaş
GMM	Olasılıksal	Esnek modelleme	Parametre tahmini zor
Fuzzy C-Means	Bulanık	Yumuşak kümeleme	Hesaplama maliyeti yüksek
Spectral	Grafik tabanlı	Kompleks yapıları yakalar	Büyük veri için maliyetli

5 Kümleme Türleri



6 Kümeleme Süreci

1. **Veri Hazırlama** – Eksik değerlerin işlenmesi, ölçekleme, özellik seçimi.
2. **Uygun Algoritma Seçimi** – Veri yapısına göre yöntem belirlenir.
3. **Model Eğitimi** – Algoritma çalıştırılır, kümeler oluşturulur.
4. **Sonuçların Değerlendirilmesi** – Görselleştirme, metrikler (Silhouette, Davies-Bouldin vb.).

7 Kümeleme ile Elde Edilen Çıktılar

- Müşteri Segmentasyonları
- Görüntü Gruplandırma
- Biyolojik Sınıflandırma
- Veri Keşfi ve Görselleştirme

8 Gerçek Dünya Örnekleri

- **E-Ticaret** – Müşteri davranışlarına göre segmentasyon
- **Sağlık** – Hastalık gruplarını belirleme
- **Görüntü İşleme** – Nesne ve görüntü sınıflandırma
- **Pazarlama** – Hedef kitle analizi ve kişiselleştirme

9 Özet

Kümeleme, etiketlenmemiş verilerdeki doğal yapıları keşfetmek, verileri anlamlı gruplara ayırmak ve birçok alanda değerli bilgiler üretmek için kullanılan güçlü bir makine öğrenimi yöntemidir. Doğru algoritma ve parametre seçimi, başarılı sonuçlar için kritik öneme sahiptir.

Doğru algoritma + Doğru veri + Doğru parametre = Anlamlı kümeler

Bilinen bazı kümeleme algoritmaları arasında K-Means Kümeleme Algoritması, Orta-lama Kaydırma Algoritması, DBSCAN Algoritması, Temel Bileşen Analizi ve Bağımsız Bileşen Analizi bulunur (Kanade, 2022).

Kümelemenin önemli bir yararı, gizli kalıpların veya veri yapılarının tespit edilmesine yardımcı olmasıdır. Bu da yazılım mühendislerini yeni keşiflere ve icatlara yönlendirmektedir. Bu tekniğin dezavantajları arasında doğru kümeleme algoritmasının seçilmesinin zorluğu yer almaktadır. Ayrıca makine öğrenimi uzmanlarının her durumda kullanılacak küme sayısına karar vermesi zor olabilmektedir (Danielkievich, 2023).

7.2.2.2. Birliktelik (Assocation)

Birliktelik kuralları, büyük veritabanlarındaki değişkenler arasındaki ilişkileri keşfetmek için kullanılır. Büyük veritabanlarındaki değişkenler arasındaki ilginç ilişkileri keşfetmeye yönelik kural tabanlı bir makine öğrenme yöntemidir (Arya, 2024).

Birliktelik (Assocation), büyük bir veri kümesinin değişkenleri arasındaki tipik ilişkilerin tanımlanması anlamına gelir. Çeşitli veri öğelerinin bağımlılığını belirler ve ilişkili değişkenleri haritalandırır. Tipik uygulamalar arasında web kullanım madenciliği ve pazar veri analizi bulunur. Birliktelik kurallarına uyan popüler algoritmalar arasında Apriori Algoritması, Eclat Algoritması ve FP-Büyüme Algoritması bulunur (Kanade, 2022).

Birliktelik (Association)

Birliktelik kuralları, büyük veritabanlarındaki değişkenler arasındaki ilişkileri keşfetmek için kullanılan kural tabanlı bir makine öğrenme yöntemidir.

Temel Tanım

Birliktelik (Association), büyük bir veri kümesinin değişkenleri arasındaki tipik ilişkilerin tanımlanması anlamına gelir. Çeşitli veri öğelerinin bağımlılığını belirler ve ilişkili değişkenleri haritalandırır.

Neden Önemlidir?

- ✓ Veri içindeki gizli ilişkileri ortaya çıkarır.
- ✓ Karar verme süreçlerine katkı sağlar.
- ✓ Pazarlama ve iş stratejilerinde değer üretir.
- ✓ Büyük veri analizi için güçlü bir araçtır.

Nasıl Çalışır?

1. **Veri Seti**
   Büyük veri kümeleri (ör. alışveriş sepeti).
2. **Sıklık Hesaplama**
   Birliktelik kuralları için destek (support) ve güven (confidence) değerleri hesaplanır.
3. **Kuralların Oluşturulması**
   Belirli eşik değerlerini sağlayan ilişkiler belirlenir.
4. **Sonuç**
   Değişkenler arasındaki anlamlı birliktelik kuralları elde edilir.

Örnek Uygulama: Alışveriş Sepeti Analizi

Alışveriş Sepeti Verisi

Müşteri ID	Ürünler
1	Süt, Ekmek, Yumurta
2	Ekmek, Peynir, Tereyağı
3	Süt, Ekmek, Peynir
4	Yumurta, Un, Şeker
5	Süt, Peynir, Tereyağı
6	Ekmek, Un, Şeker

Elde Edilen Birliktelik Kuralları

Süt ⇒ Ekmek
Destek: 0.5 | Güven: 0.8

Ekmek ⇒ Peynir
Destek: 0.4 | Güven: 0.7

Süt ⇒ Peynir
Destek: 0.3 | Güven: 0.6

Örneğin, müşterilerin %50'sinin süt ve ekmek birlikte aldığı ve bu müşterilerin %80'inin ekmek aldığında süt de aldığı keşfedilmiştir.

Temel Kavramlar

S — Destek (Support)
Birliktelik kuralının veri setinde ne kadar sıklıkla görüldüğünü gösterir.

G — Güven (*Confidence*)
Bir koşulun sağlandığında, diğer koşulun gerçekleşme olasılığını ifade eder.

L — Kaldıraç (*Lift*)
İki değişken arasındaki ilişkinin rastlantısal olup olmadığını gösterir.

Yapay Zeka ve Uygulamaları - Copyright © 2026

Popüler Algoritmalar

Apriori Algoritması

- Aday öğe kümeleri oluşturur.
- Destek eşiğine göre adayları eler.
- Yaygın olarak kullanılır.

Kullanım Alanı:
Pazar sepeti analizi, web analizi

Eclat Algoritması

- Dikey veri yapısı kullanır.
- Kesişim (intersection) işlemi ile kuralları bulur.
- Apriori'ye göre daha hızlıdır.

Kullanım Alanı:
Büyük ve yoğun veri kümeleri

FP-Büyüme Algoritması

- Veri sıkıştırma (FP-ağaç) kullanır.
- Aday üretmeden kuralları bulur.
- Çok büyük veri setlerinde etkilidir.

Kullanım Alanı:
Büyük ölçekli veri analizi

Başlıca Kullanım Alanları

Web Kullanım Madenciliği
Kullanıcı davranışlarını analiz etme

Pazar Veri Analizi
Ürün birlikteliklerini keşfetme

Pazarlama
Kampanya ve öneri sistemleri

Stok Yönetimi
Ürün yerleşimi ve satış tahmini

7.2.2.3. Boyut İndirgeme

Makine öğrenmesinde boyut indirgeme algoritmaları, yüksek boyutlu veri setlerinin boyutunu azaltarak veri setinin karmaşıklığını ve işleme zamanını azaltmayı amaçlayan tekniklerdir. Bu algoritmalar, veri analizi, görselleştirme, sınıflandırma ve kümeleme gibi uygulamalarda kullanılır.

Boyut İndirgeme (Dimensionality Reduction)

Makine öğrenmesinde boyut indirgeme algoritmaları, yüksek boyutlu veri setlerinin boyutunu azaltarak veri setinin karmaşıklığını ve işleme zamanını azaltmayı amaçlayan tekniklerdir. Bu algoritmalar, veri analizi, görselleştirme, sınıflandırma ve kümeleme gibi uygulamalarda kullanılır.

Neden Boyut İndirgeme?

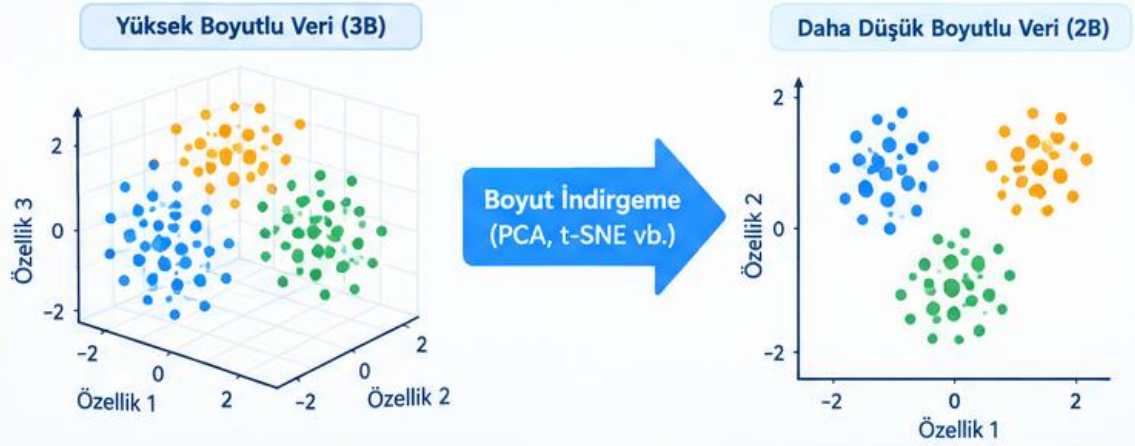
Daha hızlı işleme
Daha az özellik, daha kısa hesaplama süresi.

Karmaşıklığı azaltma
Yüksek boyutlu verilerdeki gereksiz bilgileri ortadan kaldırır.

Daha iyi görselleştirme
Veriler 2D veya 3D'de daha anlaşılır hale gelir.

Performansı artırma
Sınıflandırma ve kümeleme gibi uygulamalarda doğruluğu artırır.

Örnek: 3 Boyutlu Veriden 2 Boyuta İndirgeme



Hangi Algoritmalar Kullanılır?

PCA (Principal Component Analysis)
Doğrusal boyut indirgeme yöntemi. Varyansı en iyi açıklayan bileşenleri seçer.

t-SNE (t-Distributed Stochastic Neighbor Embedding)
Veriyi 2D/3D'de görselleştirmede çok etkilidir.

UMAP (Uniform Manifold Approximation and Projection)
Hızlı ve etkili bir boyut indirgeme tekniğidir.

LDA (Linear Discriminant Analysis)
Sınıflandırma için boyut indirgeme sağlar.

Kullanım Alanları

Veri Analizi
Büyük veri setlerinde anlamlı yapıları ortaya çıkarır.

Görselleştirme
Verileri daha anlaşılır hale getirir.

Sınıflandırma
Sınıf ayrımını kolaylaştırır.

Kümeleme
Benzer veri gruplarını daha net ayırır.

Önemli Noktalar

- Boyut indirgeme, bilgi kaybı olmadan (veya en aza indirerek) veri setini daha küçük bir boyutta temsil etmeyi hedefler.
- Hangi algoritmanın kullanılacağı, veri setinin yapısına ve amacına bağlıdır.
- Doğru uygulandığında, modelin hem performansını hem de yorumlanabilirliğini artırır.

Yapay Zeka ve Uygulamaları - Copyright © 2026

Örnek Uygulama

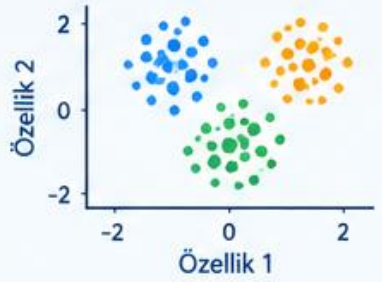
Yüksek Boyutlu Veri Seti (Örn. 50 özellik)

	Öz 1	Öz 2	Öz 3	...	Öz 50
V1	0.2	1.5	0.7	...	0.3
V2	1.1	0.3	0.6	...	1.2
V3	0.7	1.8	1.4	...	0.5
⋮	...	⋮	⋮	...	⋮

Boyut İndirgeme İşlemi

- PCA
- t-SNE
- UMAP
- LDA

Daha Düşük Boyutlu Veri

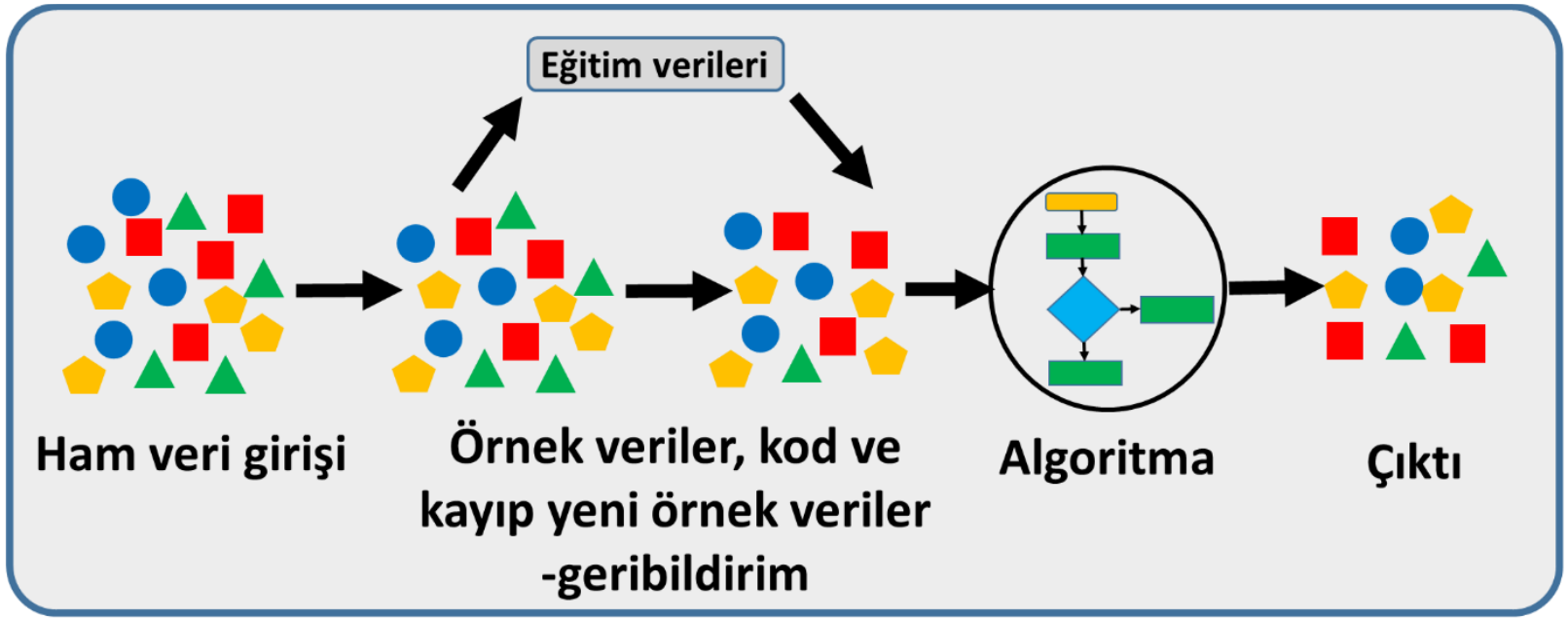


Sonuç

Daha hızlı, daha basit, daha anlaşılır ve daha iyi performans gösteren bir model.

7.2.3. Yarı Denetimli Makine Öğrenmesi

Yarı denetimli makine öğrenme algoritması, denetimli ve denetimsiz öğrenme algoritmaları arasında 'orta yol' olan başka bir makine öğrenme tekniğidir. Yarı denetimli öğrenme, eğitim verileri sırasında hem etiketli hem de etiketsiz verileri kullanır (Salah, 2024).



Şekil 7.10. Yarı denetimli makine öğrenmesi

Yapay Zeka ve Uygulamaları - Copyright © 2026

Yarı denetimli makine öğrenimi, veri noktaları arasında istenen ilişkileri anlamak için denetimli makine öğreniminden gelen sınıflandırma sürecini kullanır. Daha sonra kalan etiketlenmemiş verileri gruplandırmak için diğer denetimsiz makine öğrenimi algoritmalarından gelen kümeleme işlemini kullanır (Hutchison, 2021).

Yarı Denetimli Öğrenme Uygulamaları

- Tıbbi görüntü analizi
- Konuşma tanıma ve doğal dil işleme
- Metin sınıflandırması ve duygu analizi
- Finansta dolandırıcılık tespiti

Yarı denetimli öğrenme teknikleri, sınıflandırma, regresyon ve anormallik tespiti gibi çeşitli görevlere uygulanarak modellerin daha doğru tahminler yapmasına ve gerçek dünya senaryolarında daha iyi genelleme yapmasına olanak tanır (Tavasoli, 2024).

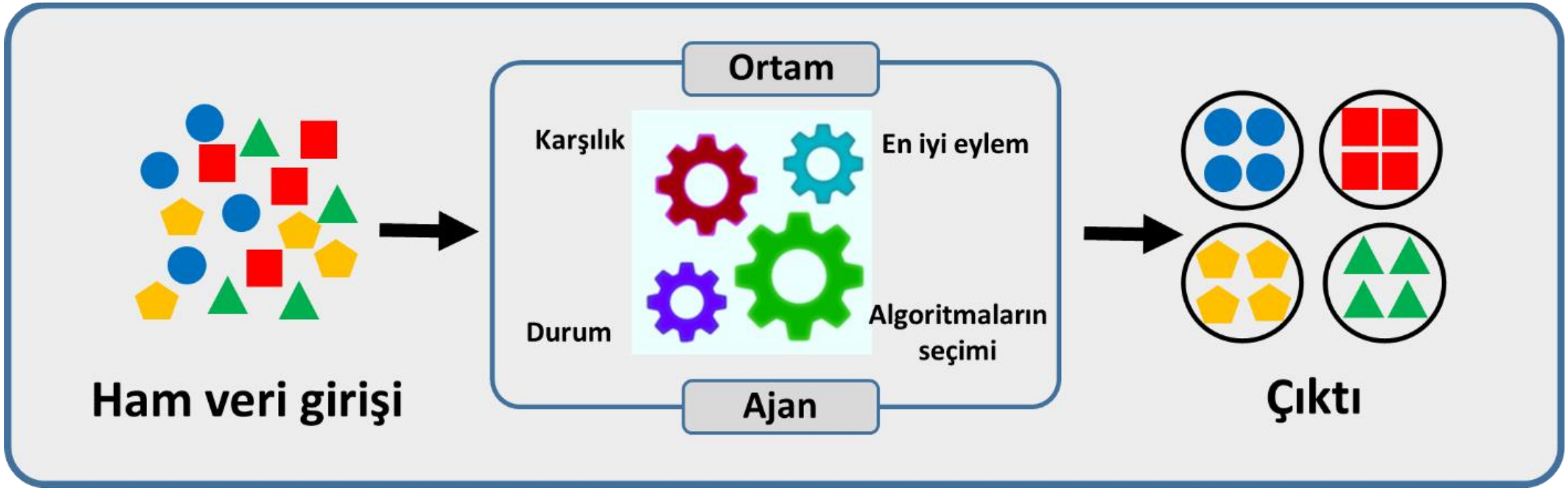
7.2.4. Takviyeli Makine Öğrenmesi

Takviyeli öğrenme, makine öğrenimi alanında önemli bir paradigmadır, çünkü bir öğrenme modeli çevresiyle etkileşime girer ve bu etkileşim sonucunda ödül veya ceza şeklinde geribildirim alarak karar verme yeteneğini geliştirir (Pardeshi, 2023).

Bu algoritma türü, oyun endüstrisinde oyun geliştirme ve robotları insan benzeri görevleri yerine getirebilecek hale getirme gibi pek çok alanda yaygın bir şekilde kullanılmaktadır (Menon, 2023). Ayrıca, yapay zeka ve robot bilimi gibi disiplinlerde de yaygın olarak araştırılmaktadır ve özellikle Go gibi karmaşık strateji oyunlarında ve satranç gibi problem çözme alanlarında başarıyla uygulanmaktadır (Pardeshi, 2023).

Takviyeli öğrenme, olumlu ve olumsuz pekiştirmeli öğrenme olmak üzere iki ana kategoriye ayrılır (Joshi, 2022). Olumlu pekiştirmeli öğrenme, bir ajanın olumlu sonuçları olan eylemleri öğrenmesini teşvik eden bir öğrenme biçimidir. Bu durumda ödül, sayısal bir değer veya sembolik bir etiket şeklinde ifade edilebilir. Öte yandan, olumsuz pekiştirmeli öğrenme, ajanın olumsuz sonuçları olan eylemlerden kaçınmasını sağlayarak öğrenmeyi teşvik eder (Joshi, 2022).

Örneğin, robotların yürümeyi öğrenmesi veya AlphaGo gibi oyun programlarının stratejik oyunlarda ustalaşması gibi uygulamalarda takviyeli öğrenme büyük bir potansiyele sahiptir. Bu algoritma, çeşitli endüstri ve disiplinlerde kullanılarak etkileyici sonuçlar elde etmek için kullanılan önemli bir araç haline gelmiştir (Suman, 2021). Takviyeli öğrenme, yapay zeka alanında ilerlemeler sağlamak ve karmaşık problemleri çözmek için önemli bir role sahiptir ve gelecekteki araştırma ve geliştirmelerde daha da yaygın olarak kullanılması beklenmektedir.



Şekil 7.11. Takviyeli makine öğrenmesi

Önemli takviyeli öğrenme algoritmalarından bazıları şunlardır:

1. Q-learning
2. Sarsa
3. Monte Carlo
4. Deep Q network

Takviyeli öğrenme, insanların deneme yanılma yöntemiyle öğrenme sürecinden ilham alarak geliştirilmiş bir makine öğrenme algoritmasıdır. Bu yaklaşım, maki-nelerin deneyimlerinden öğrenmelerini ve değişen ortamlara uyum sağlamalarını sağlayarak belirli hedeflere ulaşmalarını hedefler (Tavasoli, 2024). Takviyeli öğrenme, karmaşık karar verme problemlerini çözmek için güçlü bir teknik olarak kabul edilir.

Bu öğrenme yöntemi, makinenin farklı seçenekleri ve olasılıkları keşfetmesine olanak tanırken, her eylemin sonucunu izleyerek ve değerlendirerek en uygun olanı belirlemeye çalışır (Wakefield, 2024). Makineye deneme yanılma sürecini öğretir ve geçmiş deneyimlerden öğrenerek duruma uygun bir yaklaşım geliştirmeye başlar.

Takviyeli öğrenme, denetimli öğrenmeden farklı olarak etiketlenmiş veriler yerine yalnızca deneyimlerden öğrenme üzerine odaklanır (Kanade, 2022). Bu sayede, makinelerin çevreleriyle etkileşim halindeyken karar verme yeteneklerini geliştirmeleri ve zaman içinde daha iyi performans göstermeleri mümkün olur.

Takviyeli öğrenmenin insan deneyimine dayanan bu dinamik yaklaşımı, makine öğrenimi alanında önemli bir ilerleme sağlamış ve farklı endüstrilerde kullanım potansiyeline sahip bir algoritma olarak kabul edilmiştir.

Doğrusal Regresyon (Linear Regression)

Doğrusal regresyon, bir bağımlı değişken ile bir veya daha fazla bağımsız değişken arasındaki ilişkiyi modellemek için kullanılan temel bir makine öğrenimi tekniğidir.

Temel Bilgiler

- **Doğrusal regresyon**, veriler arasındaki ilişkiyi doğrusal bir model ile ifade eder.
- Özellikle, bir **bağımlı değişken** (ör. konut fiyatı) ile bir veya daha fazla **bağımsız değişken** (ör. evin büyüklüğü, yatak odası sayısı vb.) arasındaki ilişkiyi modellemek için kullanılır.
- Bir değişkendeki değişikliğin diğerini nasıl etkilediğini ve sonucun olumlu veya olumsuz olmasına neden olan ilişkiyi anlamaya yarar.

Örnek Problem

Konut fiyatlarını, evin büyüklüğü (m^2) ve yatak odası sayısına göre tahmin etmek.

Değişkenler

Bağımlı Değişken (Y)	Bağımsız Değişkenler (X)
Konut fiyatı (1000 TL)	• Evin büyüklüğü (m^2) • Yatak odası sayısı



Daha büyük evler ve daha fazla yatak odası genellikle daha yüksek fiyatlarla ilişkilidir.

Python Kodu

```python
# Gerekli kütüphaneleri içe aktaralım
import numpy as np
import pandas as pd
import matptotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Örnek veri seti (konut fiyatları)
data = {
    'm2': [50, 70, 90, 110, 130, 150, 170, 190, 210, 230],
    'yatak_odasi': [1, 2, 2, 3, 3, 4, 4, 5, 5, 6],
    'fiyat': [120, 180, 220, 280, 320, 380, 430, 480, 540, 600]
}
df = pd.DataFrame(data)
X = df[['m2', 'yatak_odasi']]  # Bağımsız değişkenler
y = df['fiyat']                # Bağımlı değişken

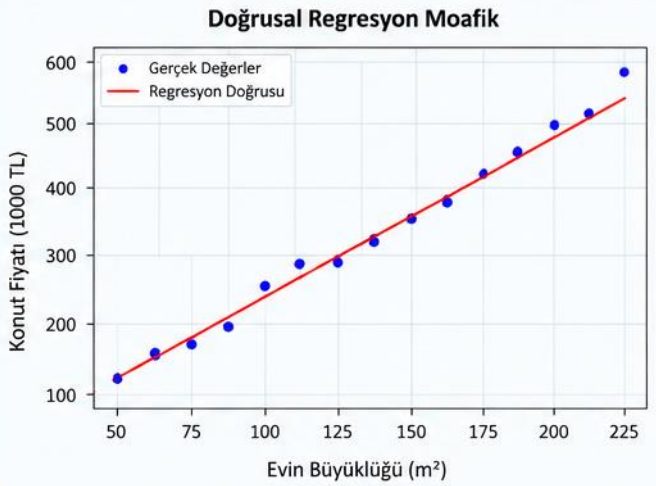
# Veriyi eğitim ve test olarak ayıralım
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42)

# Modeli oluşturalım ve eğitelim
model = LinearRegression()
model.fit(X_train, y_train)

# Tahmin yapalım ve performansı değerlendirelim
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"MSE: {mse:.2f}")
print(f"R²: {r2:.2f}")

# Sonuçları görselleştirelim (sadece m2'ye göre)
plt.scatter(df['m2'], df['fiyat'], color='blue', label='Gerçek Değerler')
plt.plot(df['m2'], model.predict(X), color='red', label='Regresyon Doğrusu')
plt.xlabel('Evin Büyüklüğü (m²)')
plt.ylabel('Konut Fiyatı (1000 TL)')
plt.title('Doğrusal Regresyon Modeli')
plt.legend()
plt.grid(True)
plt.show()
```

Sonuç: Tahmin ve Grafik



Model Sonuçları (Örnek)

MSE: 1200.35

R^2: 0.96

Yorum

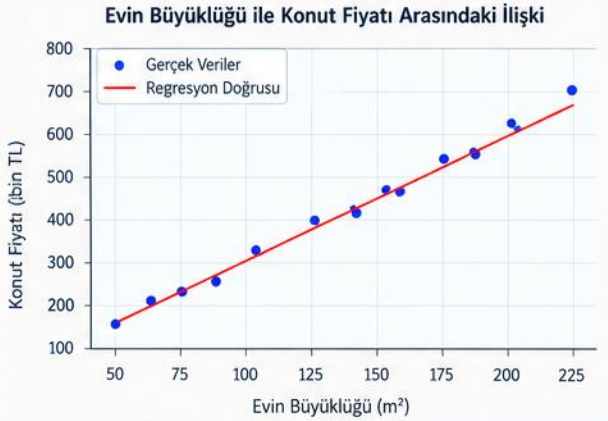
- R^2 değeri 0.96 olduğu için model, verilerdeki değişimin %96'sını açıklayabilmektedir.
- MSE değeri, tahmin hatasının büyüklüğü hakkında bilgi verir (daha düşük değerler daha iyidir).
- Grafik, ev büyüklüğü arttıkça konut fiyatlarının arttığını ve modelin verilerle iyi bir uyum sağladığını göstermektedir.

Sonuç: Doğrusal regresyon, basit yapısı ve yorumlanabilir sonuçları sayesinde birçok alanda yaygın olarak kullanılır.

Temel Kavramlar

- **Bağımlı değişken (Y):** Tahmin edilmek istenen sonuç değişkenidir. (Örn. konut fiyatı)
- **Bağımsız değişken (X):** Sonucu etkileyen değişkenlerdir. (Örn. evin büyüklüğü, yatak odası sayısı)
- **Model:** Bağımsız değişkenler ile bağımlı değişken arasındaki ilişkiyi tanımlar.
- **Amaç:** Verilerdeki ilişkiyi modelleyerek yeni gözlemler için tahmin yapmaktır.

Grafiksel Gösterim



Doğrusal regresyon, verilerdeki ilişkiyi bir doğru ile modelleyerek yeni değerler için tahmin yapar.

Doğrusal Regresyon Formülü

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

- Y : Tahmin edilmek istenen bağımlı değişken (konut fiyatı)
- X : Bağımsız değişken (evin büyüklüğü)
- β_0 : Sabit terim (intercept)
- β_1 : Eğim katsayısı (X'teki 1 birim artışın Y'deki ortalama değişimi)
- ε : Hata terimi

Birden Fazla Bağımsız Değişken (Çoklu Doğrusal Regresyon)

Konut fiyatını hem evin büyüklüğüne hem de yatak odası sayısına göre tahmin edersek, çoklu doğrusal regresyon kullanırız.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$$

Örneğin:

$$Y = 40 + 2X_1 + 20X_2$$

- X_1 : Evin büyüklüğü (m²)
- X_2 : Yatak odası sayısı
- Y : Tahmini fiyat (bin TL)

Evin büyüklüğü (X_1)	Yatak odası (X_2)	Tahmini fiyat (Y)
200 m²	3	Y = 40 + 2(200) + 20(3) = 500 **500 bin TL**

Bu örnekteki katsayılar açıklama amacıyla varsayılmıştır. Gerçek bir modelde katsayılar eğitim verilerinden öğrenilir.

Tek Değişkenli Örnek

Bir evin büyüklüğüne göre fiyatını tahmin etmek için kurulan model:

$$Y = 50 + 3X$$



Burada:
- Y: Konut fiyatı (bin TL)
- X: Evin büyüklüğü (m²)

Ev büyüklüğü (X)	Hesaplama	Tahmini fiyat (Y)
50 m²	50 + 3 × 50	200 bin TL
100 m²	50 + 3 × 100	350 bin TL
150 m²	50 + 3 × 150	500 bin TL
200 m²	50 + 3 × 200	650 bin TL

Örnek hesaplama: 200 m² için
Y = 50 + 3(200) = 650 → **Tahmini fiyat: 650.000 TL**

Özet

- Doğrusal regresyon, bağımsız değişkenler ile bağımlı değişken arasındaki ilişkiyi modelleyen basit ve etkili bir yöntemdir.
- Formülü: $Y = \beta_0 + \beta_1 X + \varepsilon$
- Tek değişkenle basit, birden fazla değişkenle çoklu regresyon olarak kullanılır.
- Konut fiyatı, tahmin, satış, talep, vb. birçok alanda uygulanabilir.
- Modelin başarısı, verilerin kalitesi ve seçilen değişkenlere bağlıdır.

Doğrusal regresyon, verilerdeki ilişkiyi anlamanızı ve geleceğe yönelik güvenilir tahminler yapmanızı sağlar.

Yapay Zeka ve Uygulamaları - Copyright © 2026

Lojistik Regresyon (Logistic Regression)

Nedir?

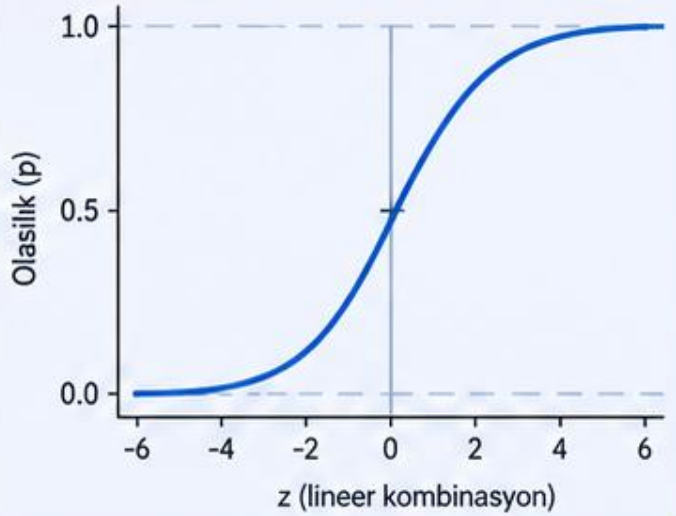
Lojistik regresyon, sağlanan önceki verilere dayanarak bir olayın meydana gelme olasılığını tahmin etmeye odaklanan bir makine öğrenimi tekniğidir (Wakefield, 2024). Bu algoritma, sınıflandırma problemlerinde kullanılır ve özellikle bir veri noktasını belirli bir sınıfa (örneğin, 0 veya 1) atamak için idealdir. Lojistik regresyon, çıktıyı 0 ile 1 arasında olan olasılık değerleri olarak verir ve genellikle sigmoid fonksiyonu kullanılarak modelin çıktısı hesaplanır.

Özellikleri

- İkili sınıflandırma problemlerinde kullanılır.
- Çıktıyı 0–1 arasında olasılık değerleri olarak verir.
- Sigmoid (lojistik) fonksiyonu ile çalışır.
- Tıp, finans, pazarlama gibi birçok alanda yaygın olarak kullanılır.

Sigmoid Fonksiyonu

$$p = \frac{1}{1 + e^{-z}}$$



Kullanım Alanları

- Kredi Puanlama
- Kanser Tespiti
- Coğrafi Görüntü İşleme
- Elyazısı Tanıma
- Görüntü Segmentasyonu ve Kategorizasyonu
- Pazarlama Kampanyalarının Başarı Oranlarının Ölçülmesi
- Belirli Bir Ürünün Gelirini Tahmin Etmek
- Deprem Tahmini

Yapay Zeka ve Uygulamaları - Copyright © 2026

Python ile Lojistik Regresyon Örneği

```python
# Gerekli kütüphaneleri içe aktaralım
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Örnek veri seti (Müşteri bilgileri ve satın alma durumu)
data = {
    'yas': [25, 34, 45, 23, 52, 41, 36, 29, 48, 31],
    'gelir': [50000, 62000, 80000, 42000, 95000, 72000, 68000, 54000, 88000, 60000],
    'kredi_puani': [650, 700, 780, 620, 800, 730, 710, 660, 760, 690],
    'satın_alma': [0, 0, 1, 0, 1, 1, 0, 0, 1, 0]
}
df = pd.DataFrame(data)

# Özellikler ve hedef değişkeni ayıralım
X = df[['yas', 'gelir', 'kredi_puani']]
y = df['satın_alma']

# Eğitim ve test setlerine ayıralım
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Modeli oluşturalım ve eğitelim
model = LogisticRegression()
model.fit(X_train, y_train)

# Tahmin yapalım
y_pred = model.predict(X_test)
y_prob = model.predict_proba(X_test)[:, 1]   # Satın alma olasılıkları

# Sonuçları değerlendirelim
print("Doğruluk (Accuracy):", accuracy_score(y_test, y_pred))
print("\nSınıflandırma Raporu:\n", classification_report(y_test, y_pred))
print("\nKarışıklık Matrisi:\n", confusion_matrix(y_test, y_pred))
```

Model Sonuçları

Doğruluk (Accuracy):
0.90

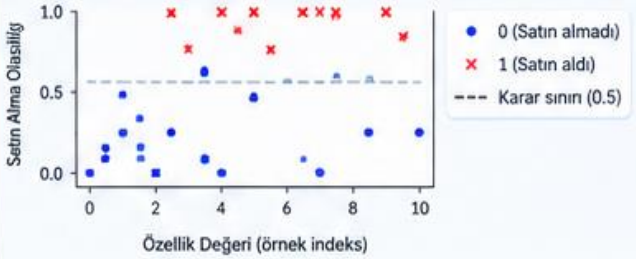
Sınıflandırma Raporu

	precision	recall	f1-score	support
0	0.89	0.91	0.90	11
1	0.91	0.89	0.90	9
accuracy			0.90	20
macro avg	0.90	0.90	0.90	20
weighted avg	0.90	0.90	0.90	20

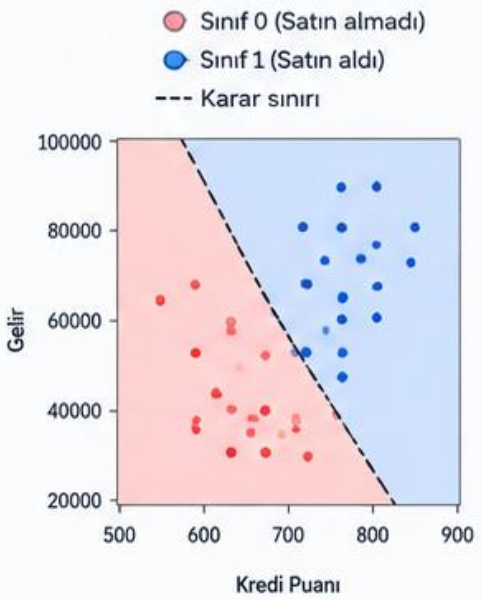
Karışıklık Matrisi

	Tahmin 0	Tahmin 1
Gerçek 0	10	1
Gerçek 1	1	8

Tahmin Olasılıkları (Test Seti)



Karar Sınırı ve Olasılıklar



Not

Gerçek projelerde daha büyük veri setleri, daha fazla özellik ve hiperparametre ayarlamaları ile model performansı artırılabilir.

Yapay Zeka ve Uygulamaları - Copyright © 2026

Karar Ağacı ve Rastgele Ormanlar (Random Forest)

Karar Ağacı

- Karar ağacı, hem sınıflandırma hem de regresyon görevleri için kullanılabilen denetimli öğrenme algoritmasıdır (Reilly, 2023).
- Verileri belirli koşullara (bölme kuralları) dayalı olarak farklı gruplara ayırır. Her düğüm bir değişkene ilişkin testi temsil eder, her dal testin sonucudur (Wakefield, 2024).
- En büyük avantajı **yorumlanabilir** olmasıdır. Karar ağacı **görselleştirilebilir** ve kararların nasıl verildiği takip edilebilir (Reilly, 2023).

Rastgele Ormanlar

- Rastgele orman, birden fazla karar ağacını kullanarak tahmin yapan bir topluluk (ensemble) öğrenme algoritmasıdır (Reilly, 2023).
- Her ağaç farklı veri seti ve rastgele değişkenlerle oluşturulur. Çoğunluk oylaması ile nihai tahmin yapılır.
- Aşırı uyumu azaltır ve daha doğru, **ölçeklenebilir** sonuçlar verir. Ancak yorumlanması karar ağaçlarına göre daha zordur (Wahab, 2023).

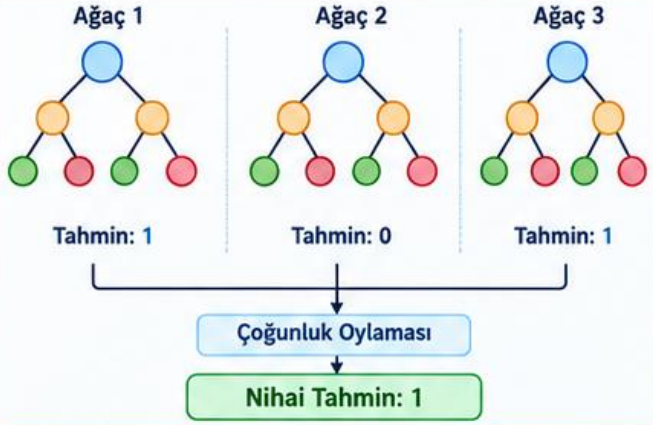
Kullanım Alanları

- Sınıflandırma (başarı durumu, risk analizi)
- Regresyon (süre, fiyat tahmini)
- Sağlık (hastalık tahmini)
- Finans (kredi skoru)
- Pazarlama (müşteri analizi)
- Eğitim (öğrenci başarısı)
- Doğal kaynaklar (çevre analizi)

Tek Bir Karar Ağacı Örneği



Rastgele Orman (Birden Fazla Ağaç)



Avantajları

- Yüksek doğruluk
- Aşırı uyuma karşı dayanıklı
- Hem sınıflandırma hem regresyon
- Ölçeklenebilir
- Özellik önemini belirleyebilir

Dezavantajları

- Yorumlanması daha zordur
- Daha fazla hesaplama kaynağı gerektirir
- Büyük veri setlerinde daha yavaş olabilir

```python
# Gerekli kütüphaneler
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Örnek veri seti (öğrenci notları ve başarı durumu)
data = {
    'Matematik': [85, 72, 60, 78, 92, 55, 68, 88, 76, 64],
    'Fizik': [80, 68, 58, 75, 90, 50, 62, 85, 70, 60],
    'Basari': [1, 1, 0, 1, 1, 0, 0, 1, 1, 0]
}
df = pd.DataFrame(data)

# Eğitim ve test setlerine ayırma
X = df[['Matematik', 'Fizik']]
y = df['Basari']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Karar Ağacı Modeli
tree_model = DecisionTreeClassifier(max_depth=3, random_state=42)
tree_model.fit(X_train, y_train)
tree_pred = tree_model.predict(X_test)
print("Karar Ağacı Doğruluğu:", accuracy_score(y_test, tree_pred))

# Rastgele Orman Modeli
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
rf_pred = rf_model.predict(X_test)
print("Rastgele Orman Doğruluğu:", accuracy_score(y_test, rf_pred))
```

Örnek Veri Seti

Matematik	Fizik	Başarı
85	80	1
72	68	1
60	58	0
78	75	1
92	90	1
55	50	0
68	62	0
88	85	1
76	70	1
64	60	0

Sınıflar

1 → Başarılı

0 → Başarısız

Model Performansı

Doğruluk (Accuracy)



Karışıklık Matrisi (Karar Ağacı)

	Tahmin	0	1
Gerçek	0	2	1
	1	1	6

Karışıklık Matrisi (Rastgele Orman)

	Tahmin	0	1
Gerçek	0	3	0
	1	0	7

Sonuç

Karar ağacı, verilerin nasıl bölündüğünü açıkça göstererek yorumlanabilir bir model sunar.

Rastgele orman ise birden fazla ağacı birleştirerek daha yüksek doğruluk sağlar ve aşırı uyumu azaltır.

Bu örnekte, öğrenci notlarına dayanarak başarı durumunu tahmin etmek için her iki yöntem de etkili bir şekilde kullanılabilir.

"Doğru model, doğru kararlar için güçlü bir araçtır."

Naive Bayes

Naive Bayes Nedir?

- Naive Bayes, tüm özelliklerin birbirinden bağımsız olduğunu varsayan Bayes teoremini temel alan olasılıksal bir makine öğrenme modelidir.
- Bu algoritma, bir öğenin belirli bir sınıfa ait olma ve en yüksek olasılığa sahip sınıfa atanma olasılığını tahmin eder (Wahab, 2023).
- Modelin oluşturulması kolaydır, büyük veri kümeleri için kullanışlıdır. Basittir ve son derece karmaşık sınıflandırma yöntemlerinden bile daha iyi performans gösterebilir (Tavasoli, 2024).

Nasıl Çalışır?

Bayes Teoremi

$$P(C \mid X) = \frac{P(X \mid C) \cdot P(C)}{P(X)}$$

Burada:

C	: Sınıf (ör. 0 veya 1)
X	: Özellikler (ör, yaş, gelir, vb.)
$P(C \mid X)$	: X verildiğinde C sınıfının olasılığı
$P(X \mid C)$	: C sınıfı verildiğinde X'in olasılığı
$P(C)$	: Sınıfın önsel olasılığı
$P(X)$	: Tüm olasılık (normalizayson için)

Özellikleri ve Avantajları

- Tüm özelliklerin birbirinden bağımsız olduğu varsayımına dayanır.
- Modeli oluşturmak ve eğitmek kolaydır.
- Büyük veri kümeleri için idealdir.
- Hızlıdır ve düşük hesaplama maliyeti vardır.
- Çoğu zaman karmaşık yöntemlerden daha iyi performans gösterebilir.

Örnek: Sınıflandırma

Öğrencilerin sınav sonuçlarına göre başarı durumunu tahmin etme

Öğrenci	Matematik	Fizik	Sınıf
Ali	85	80	1
Ayşe	72	68	1
Mehmet	60	58	0
Zeynep	78	75	1
Emre	55	50	0
Elif	68	62	0



Olasılık Hesaplamaları

1. Sınıf Önsel Olasılıkları

$$P(1) = \frac{3}{6} = 0.50 \quad P(0) = \frac{3}{6} = 0.50$$

2. Koşullu Olasılıklar (Örnek: Matematik)

Başarılı (1)	Başarısız (0)
P(Matematik | 1)	P(Matematik | 0)
85: 1/3 = 0.33	60: 1/3 = 0.33
72: 1/3 = 0.33	55: 1/3 = 0.33
78: 1/3 = 0.33	68: 1/3 = 0.33



(Gerçek uygulamada Gauss dağılımı kullanılarak sürekli değerler için olasılıklar hesaplanır.)

Yeni Veri İçin Tahmin

Örnek: Matematik = 75, Fizik = 70

Adım 1: Her sınıf için olasılıkları hesapla

$$P(1 \mid X) \propto P(75 \mid 1) \cdot P(70 \mid 1) \cdot P(1) \rightarrow 0.42$$

$$P(0 \mid X) \propto P(75 \mid 0) \cdot P(70 \mid 0) \cdot P(0) \rightarrow 0.18$$

Adım 2: En yüksek olasılığa sahip sınıfı seç

Tahmin: Başarılı (1)

```python
# Naive Bayes sınıflandırma örneği
import pandas as pd
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report

# Örnek veri seti
data = {
    'Matematik': [85, 72, 60, 78, 55, 68],
    'Fizik': [80, 68, 58, 75, 50, 62],
    'Durum': [1, 1, 0, 1, 0, 0]
}
df = pd.DataFrame(data)

# Veriyi ayır (eğitim ve test)
X = df[['Matematik', 'Fizik']]
y = df['Durum']
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42)

# Modeli oluştur ve eğit
model = GaussianNB()
model.fit(X_train, y_train)

# Tahmin yap ve değerlendir
y_pred = model.predict(X_test)
print("Doğruluk (Accuracy):", accuracy_score(y_test, y_pred))
print("Sınıflandırma Raporu:\n", classification_report(y_test, y_pred))
```

Çıktı (örnek)

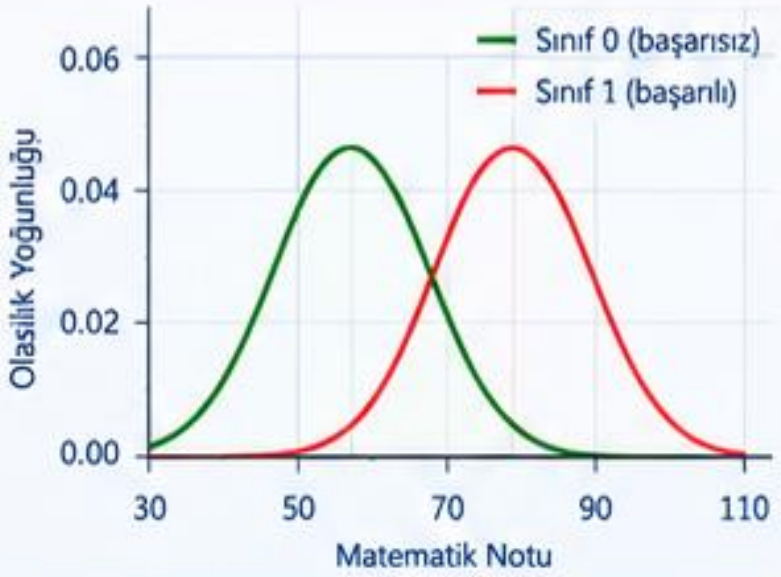
Doğruluk (Accuracy): 0.83

Sınıflandırma Raporu:

	precision	recall	f1-score	support
0	0.75	1.00	0.86	3
1	1.00	0.67	0.80	3
accuracy			0.83	6
macro avg	0.88	0.83	0.83	6
weighted avg	0.88	0.83	0.83	6

Gauss Dağılımı (Sürekli Veriler)

Özelliklerin normal dağılım gösterdiği varsayılır.



Avantajlar

- ✔ Basit ve hızlıdır.
- ✔ Küçük ve büyük veri kümelerinde etkilidir.
- ✔ Eksik verilerle başa çıkabilir.

Uygulama Alanları

- Kredi Puanlama
- Hastalık Teşhisi (Tıbbi Görüntüleme)
- Metin Sınıflandırma (Spam Tespiti)
- Duygu Analizi (Sosyal Medya)
- Müşteri Davranışı Analizi
- Doküman Sınıflandırma
- Yüz Tanıma
- Dolandırıcılık Tespiti

> *"Naive Bayes, basit varsayımlara rağmen çoğu zaman şaşırtıcı derecede iyi sonuçlar verir."*

K-Means Kümeleme (K-Means Clustering)



K-Means, benzer özelliklere sahip verileri aynı grupta (**kümelerde**) toplayan, **denetimsiz** bir makine öğrenmesi algoritmaıdır. Verilerdeki doğal yapıları ve desenleri keşfetmek için kullanılır (Wahab, 2023).

Temel Kavramlar

- **K-Means,** verileri **K** adet kümeye ayırır.
- Her küme, bir **merkez noktası (centroid)** ile temsil edilir.
- **Amaç,** her veri noktasının kendi kümesine olan uzaklığını en aza indirmektir.
- Genellikle sayısal verilerde kullanılır.

Kısaca: K-Means, verileri benzerliklerine göre kümelere ayıran, merkezleri iteratif olarak güncelleyen bir algortmadır.

Matematiksel Formül

K-Means, her kümede bulunan veri noktalarının merkezine olan uzaklıkların toplamını minimize eder.

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2$$

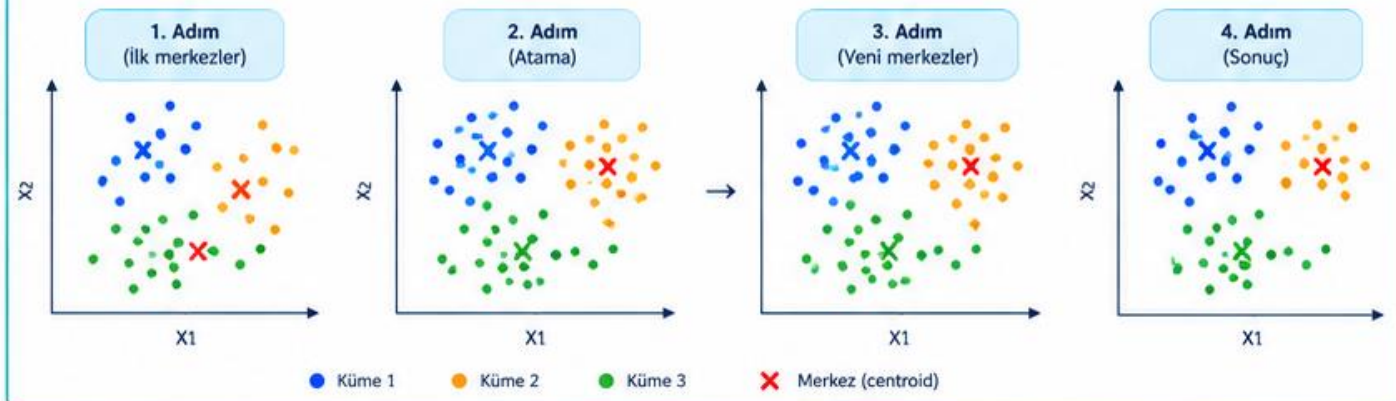
Burada:

- J : Toplam hata (küme içi kareler toplamı)
- x : Veri noktası
- μ_i : i. kümenin merkezi (centroid)
- C_i : i. kümedeki veri noktaları
- k : Küme sayısı

Nasıl Çalışır? (Adım Adım)

1. Küme sayısı (k) belirlenir.
2. Rastgele k adet merkez noktası (centroid) seçilir.
3. Her veri noktası, en yakın merkeze atanır.
4. Her kümenin yeni merkezi, o kümedeki noktaların ortalaması alınarak hesaplanır.
5. Merkezler değişmeyene kadar 3 ve 4. adımlar tekrarlanır.

Görsel Örnek: 2 Boyutlu Veri Seti



Örnek Veri Seti

Müşteri No	Maaş (Bin TL)	Harcama (Bin TL)
1	30	15
2	32	17
3	28	16
4	75	60
5	80	65
6	72	58
7	15	10
8	18	12
9	20	14

Python Kodu

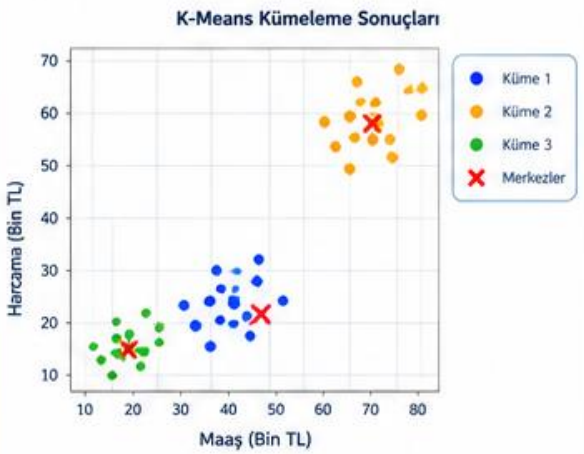
```python
# Gerekli kütüphaneleri içe aktaralım
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

# Örnek veri seti (Maaş ve Harcama)
data = np.array([
    [30, 15], [32, 17], [28, 16],
    [75, 60], [80, 65], [72, 58],
    [15, 10], [18, 12], [20, 14]
])

# K-Means modeli
kmeans = KMeans(n_clusters=3, random_state=42)
labels = kmeans.fit_predict(data)
centers = kmeans.cluster_centers_

# Sonuçları görselleştirelim
plt.figure(figsize=(8, 6))
plt.scatter(data[:, 0], data[:, 1], c=labels, cmap='viridis', s=100)
plt.scatter(centers[:, 0], centers[:, 1], c='red', marker='X', s=200, label='Merkezler')
plt.xlabel('Maaş (Bin TL)')
plt.ylabel('Harcama (Bin TL)')
plt.title('K-Means Kümeleme Sonuçları')
plt.legend()
plt.show()
```

Kümeleme Sonuçları



Bu örnekte, 3 küme (k=3) belirlenmiştir. Veriler, maaş ve harcama özelliklerine göre 3 farklı grupta toplanmıştır.

Kullanım Alanları

- Müşteri segmentasyonu (ör. pazarlama)
- Görüntü işleme (ör. nesne sınıflandırma)
- Web analizi (ör. kullanıcı davranışları)
- Sağlık verileri (ör. hasta grupları)
- Veri madenciliği ve keşifsel analiz

Avantajları ve Dikkat Edilmesi Gerekenler

Avantajları

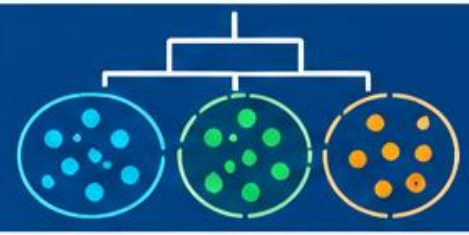
- Kolay uygulanır ve hızlıdır.
- Büyük veri setlerinde etkilidir.
- Denetimsiz öğrenmede yaygın olarak kullanılır.

Dikkat Edilmesi Gerekenler

- K küme sayısı önceden belirlenmelidir.
- Başlangıç noktaları sonucu etkileyebilir.
- Farklı büyüklükteki ve şekilli kümeler için her zaman ideal olmayabilir.

Hiyerarşik Kümeleme (Hierarchical Clustering)

Hiyerarşik kümeleme, verileri benzerliklerine göre hiyerarşik bir yapı (ağaç) şeklinde gruplandıran denetimsiz bir makine öğrenimi tekniğidir. Veriler, tek bir küme olarak başlar ve adım adım daha küçük kümelere ayrılır veya kümeler birleştirilir. Sonuçta, kümeler arasındaki ilişkiyi gösteren bir dendrogram (ağaç diyagramı) elde edilir.



Temel Kavramlar

Küme
Benzer özelliklere sahip veri noktalarının oluşturduğu gruptur.

Dendrogram
Kümeler arasındaki hiyerarşik ilişkiyi gösteren ağaç diyagramıdır.

Uzaklık Ölçüsü
Veriler arasındaki benzerliği belirlemek için kullanılır. (Öklid, Manhattan, vb.)

Bağlantı Yöntemleri
Kümeler arası mesafeyi hesaplamak için kullanılan yöntemlerdir. (Tek bağlantı, Tam bağlantı, Ortalama, Ward vb.)

Matematiksel Temel

Hiyerarşik kümeleme, genellikle bir **uzaklık matrisi** ve **bağlantı yöntemi** kullanılarak çalışır.

$$d(A,B) = \sqrt{\sum_{i=1}^{n} (x_{iA} - x_{iB})^2}$$

Burada;

- $d(A,B)$: A ve B **kümeleri arasındaki uzaklık**
- x_{iA} : A kümesindeki i. veri noktası
- x_{iB} : B kümesindeki i. veri noktası
- n : Özellik sayısı

Bağlantı yöntemi, kümeler arasındaki mesafeyi hesaplamak için farklı kriterler kullanır.

- Tek bağlantı (Single Linkage)
- Tam bağlantı (Complete Linkage)
- Ortalama bağlantı (Average Linkage)
- Ward yöntemi (Varyans minimizasyonu)

Nasıl Çalışır? (Adım Adım)

1. Tüm veri noktaları ayrı birer küme olarak başlar.
2. En yakın iki küme belirlenir (uzaklık matrisi kullanılır).
3. Bu kümeler birleştirilir.
4. Yeni küme için uzaklıklar yeniden hesaplanır.
5. İşlem, tek bir küme kalana kadar tekralanır.

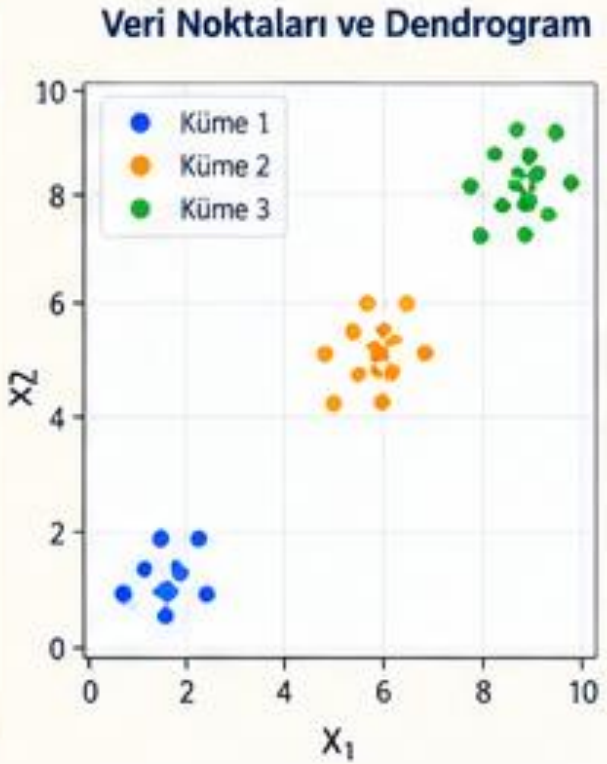


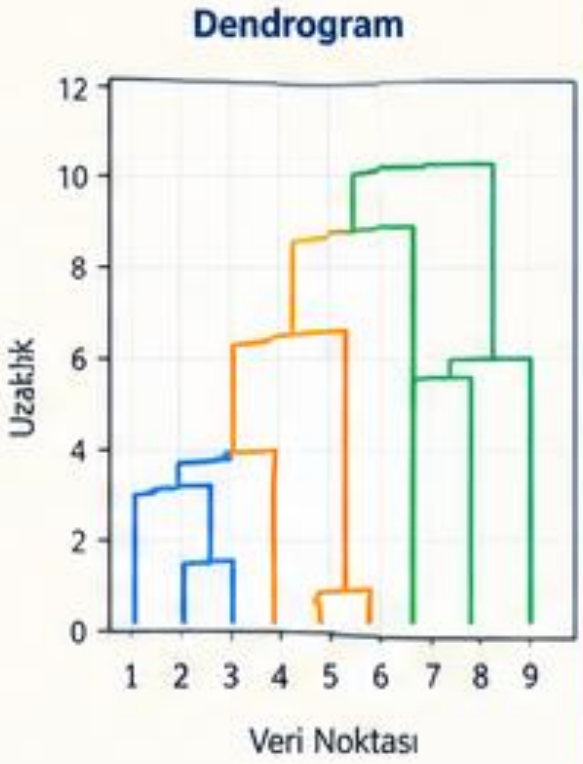
Yapay Zeka ve Uygulamaları - Copyright © 2026

Örnek Veri Seti

Aşağıdaki tabloda 2 özellikli (X1, X2) basit bir veri seti ve bu verilere ait hiyerarşik kümeleme sonucu gösterilmiştir.

No	X_1	X_2
1	1.2	1.0
2	1.5	1.2
3	1.8	0.8
4	5.0	5.2
5	5.5	5.8
6	6.0	5.1
7	8.0	8.5
8	8.5	9.0
9	9.0	8.2





Yapay Zeka ve Uygulamaları - Copyright © 2026

Python Kodu

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import linkage, dendrogram
from sklearn.preprocenssig import StandardScaler

# Örnek veri seti
data = pd.DataFrame({
    'X1': [1.2, 1.5, 1.8, 5.0, 5.5, 6.0, 8.0, 8.5, 9.0],
    'X2': [1.0, 1.2, 0.8, 5.2, 5.8, 5.1, 8.5, 9.0, 8.2]
})
# Standartlaştırma
scaler = StandardScaler()
X_scaled = scaler.fit_transform(data)

# Hiyerarşik kümeleme (Ward yöntemi)
Z = linkage(X_scaled, method='ward')

# Dendrogram
plt.figure(figsize=(12, 6))
dendrogram(Z, labels=data.index+1, color_threshold=5)
plt.title('Hiyerarşik Kümeleme Dendrogramı')
plt.xlabel('Veri Noktası')
plt.ylabel('Uzaklik')
plt.show()
```

Sonuç ve Yorum

- Dendrogram, 3 ana küme olduğunu göstermektedir.
- 1-3. noktalar bir küme, 4-6. noktalar ikinci bir küme, 7-9. noktalar ise üçüncü bir kümdir.
- Hiyerarşik kümeleme, küme sayısı önceden bilinmediğinde veya küme yapısının detaylı incelenmesi gerektiğinde avantajlıdır.

Kullanım Alanları

- **Pazarlama:** Müşteri segmentasyonu
- **Biyoinformatik:** Genetik veri analizi
- **Sosyal Bilimler:** Benzer davranış grupları
- **Görüntü İşleme:** Görüntü sınıflandırma
- **Veri Keşfi:** Veri yapısının görselleştirilmesi

Avantajları ve Dikkat Edilmesi Gerekenler

Avantajları

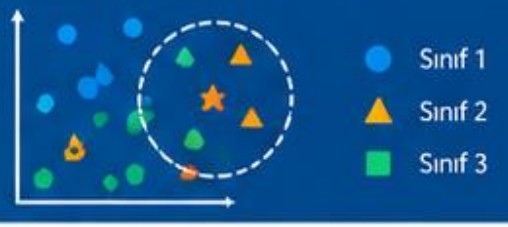
- Küme sayısı önceden belirlenmez.
- Dendrogram ile görselleştirme sağlar.
- Farklı uzaklık ve bağlantı yöntemleri kullanılabilir.

Dikkat Edilmesi Gerekenler

- Büyük veri setlerinde hesaplama maliyeti yüksektir.
- Aykırı değerlere (outlier) duyarlıdır.
- Sonuçlar, seçilen bağlantı yöntemine bağlı olarak değişebilir.

K-En Yakın Komşu (K-Nearest Neighbor - KNN)

KNN, yeni bir veri noktasının sınıfını veya **sayısal değerini**, veri setindeki **en yakın komşularına** bakarak tahmin eden, basit ama etkili bir makine öğrenmesi algoritmasıdır.



Temel Kavramlar

K (Komşu Sayısı)
Yeni bir veri noktasının sınıfı, en yakın K komşusunun çoğunluğuna göre belirlenir.

Uzaklık Ölçüsü
Genellikle Öklid uzaklığı kullanılır. (Diğer ölçüler de kullanılabilir.)

Sınıflandırma / Regresyon
KNN hem sınıflandırma (kategorik) hem de regresyon (sayısal problemlerinde kullanılabilir.

Çalışma Mantığı
Yeni veri noktası, eğitim verisindeki en yakın K komşusu araştırılır ve çoğunluk sınıfı (veya ortalama) tahmin edilir.

Nasıl Çalışır? (Adım Adım)

1. Eğitim verileri ve yeni veri noktası alınır.
2. Yeni noktanın, tüm eğitim verilerine olan uzaklığı hesaplanır.
3. En yakın K komşu belirlenir.
4. K komşuların sınıf etiketleri analiz edilir (çoğunluk oylanır).
5. Yeni noktanın sınıfı tahmin edilir.

Yeni veri noktası → Uzaklık hesapla → En yakın K komşu → Çoğunluk oylaması → Tahmin

Uzaklık Formülü

En yaygın kullanılan uzaklık ölçüsü Öklid uzaklığıdır:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Burada;

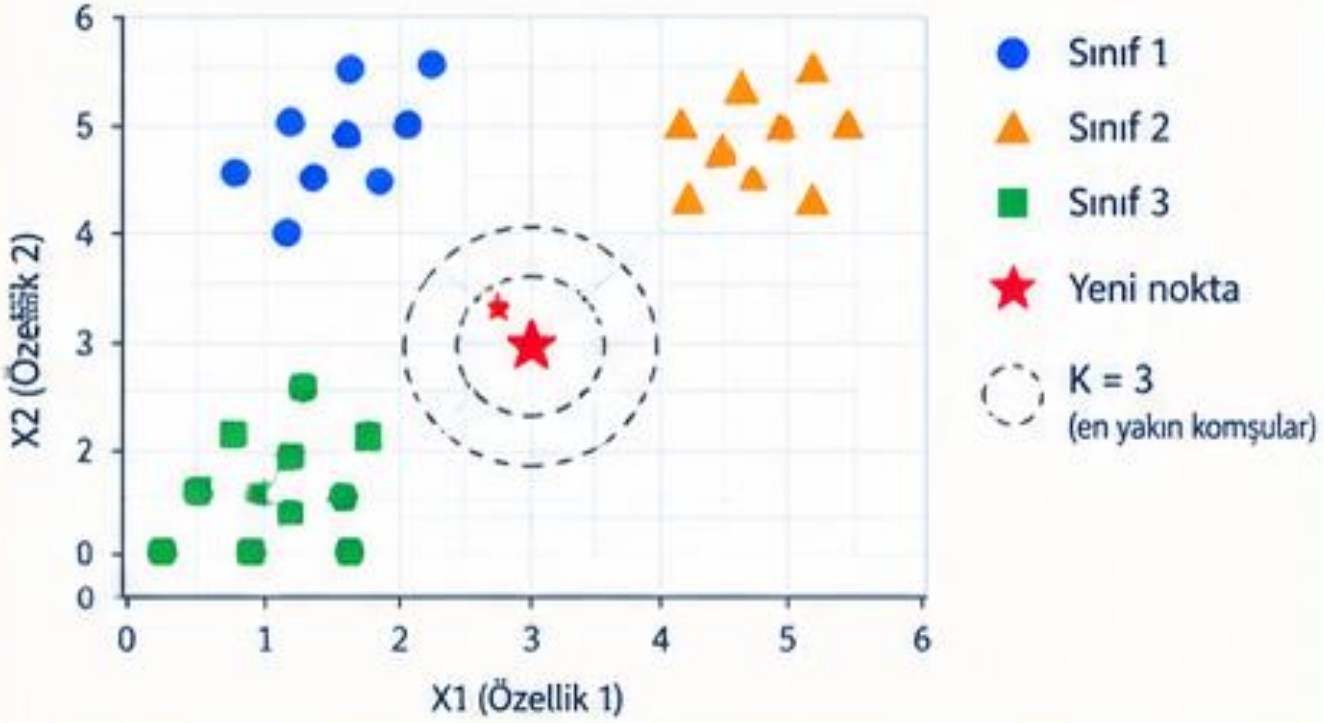
- x : Yeni veri noktası
- y : Eğitim veri noktası
- n : Özellik sayısı

Örnek:
İki nokta arasındaki Öklid uzaklığı:

$$d = \sqrt{(2-1)^2 + (3-1)^2} = \sqrt{5} \approx 2.24$$

Yapay Zeka ve Uygulamaları - Copyright © 2026

Görsel Örnek: 2 Boyutlu Veri Seti (K = 3)



K = 3 için en yakın üç komşu incelendiğinde, çoğunluk Sınıf 1 olduğu için yeni nokta Sınıf 1 olarak tahmin edilir.

Python Kodu (Sınıflandırma Örneği)

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Örnek veri seti
X = np.array([[1, 1], [1, 2], [2, 1], [2, 2],
              [5, 5], [5, 6], [6, 5], [6, 6],
              [3, 5], [3, 6], [4, 5], [4, 6]])
y = np.array([0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2])

# Model oluşturma
knn = KNeighborsClassifier(n_neighbors=3)

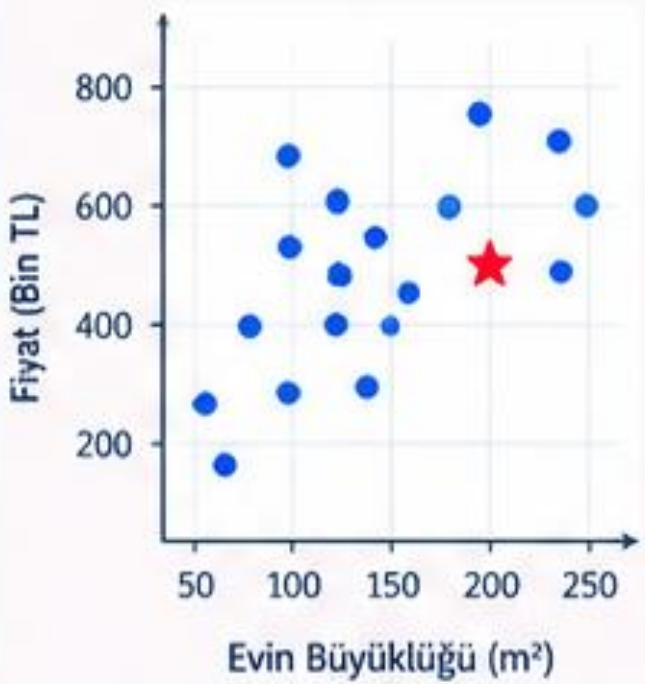
# Eğitim ve tahmin
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=42)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)

# Doğruluk oranı
print(f"Doğruluk: {accuracy_score(y_test, y_pred):.2f}")
```

Regresyon Örneği

KNN, sayısal değer tahmini için de kullanılabilir.

Örneğin, ev fiyatı tahmini:



- Eğitim verileri
- Tahmin edilecek nokta

Yeni noktanın en yakın K komşusunun fiyat ortalaması, tahmin değerini verir.

Kullanım Alanları



E-ticaret
(ürün önerileri)



Sağlık
(hastalık tahmini)



Finans
(kredi skoru)



Güvenlik
(sahtekarlık tespiti)



Pazarlama
(müşteri segmentasyonu)

Avantajları

- Basit ve anlaşılırdır.
- Eğitim süreci yoktur (veri saklar).
- Hem sınıflandırma hem regresyon problemlerinde kullanılabilir.
- Farklı veri türleriyle (sayısal/kategorik) çalışabilir.
- Küçük ve orta ölçekli veri setlerinde etkilidir.

Dezavantajları

- Büyük veri setlerinde tahmin maliyeti yüksektir.
- Uzaklık ölçüsüne ve K değerine duyarlıdır.
- Özelliklerin ölçeklenmesi gerekebilir (standartlaştırma).
- Yüksek boyutlu veri setlerinde performansı düşer ("boyutlar laneti").



Destek Vektör Makineleri (Support Vector Machine - SVM)

SVM, sınıflandırma ve regresyon problemlerinde en iyi ayırıcı hiper-düzlemi bulmak için kullanılan güçlü bir makine öğrenmesi algoritmasıdır. Özellikle yüksek boyutlu verilerde ve karmaşık karar sınırlarında etkilidir (Wahab, 2023; Reilly, 2023).

- Yüksek doğruluk
- Küçük örneklemlerde etkili
- Yüksek boyutlu verilerde başarılı

Temel Kavramlar

Hiper-düzlem
Sınıfları ayıran en uygun düzlemdir.

Destek Vektörleri
Karar sınırına en yakın olan veri noktalarıdır.

Marj (Margin)
Sınıfları ayıran hiper-düzlem ile en yakın noktalar arasındaki mesafedir.

Kernel (Çekirdek) Fonksiyonu
Verileri daha yüksek boyutlu bir uzaya taşıyarak doğrusal olmayan ayrımları mümkün kılar.

Matematiksel Formül

SVM, marjı maksimize eden bir hiper-düzlem bulur.

$$w \cdot x + b = 0$$

x : Özellik vektörü
w : Ağırlık vektörü
b : Bias (sapma)

Sınıflandırma için karar fonksiyonu:

$$f(x) = \text{sign}(w \cdot x + b)$$

Sınıflar arasındaki marjın en büyük olduğu hiper-düzlem seçilir.

Marj genişliği:

$$\frac{2}{\|w\|}$$

Nasıl Çalışır? (Adım Adım)

1. Veri seti hazırlanır ve etiketlenir.
2. Marjı maksimize eden en iyi hiper-düzlem bulunur.
3. Destek vektörleri belirlenir.
4. Yeni veriler, karar fonksiyonu ile sınıflandırılır (veya regresyon yapılır).

Örnek: Karar Fonksiyonu

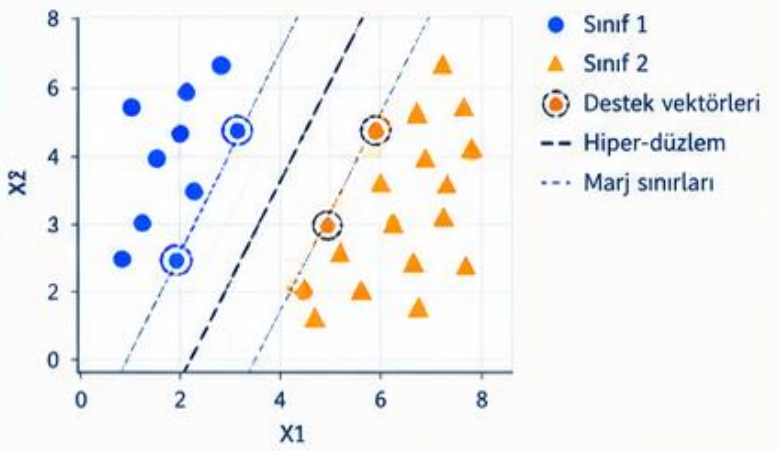
Bir noktanın hangi sınıfa ait olduğunu belirlemek için:

$$f(x) = \text{sign}(w \cdot x + b)$$

$> 0 \rightarrow$ **Sınıf 1**
$< 0 \rightarrow$ **Sınıf 2**

Yapay Zeka ve Uygulamaları - Copyright © 2026

Sınıflandırma Örneği (2 Boyutlu Veri)



SVM, sınıfları en iyi ayıran hiper-düzlemi bulur ve bu düzleme en yakın noktaları (destek vektörleri) kullanır.

Python Kodu (Sınıflandırma - Linear SVM)

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Örnek veri seti
X = np.array([[1, 2], [2, 3], [3, 3], [4, 4], [5, 5],
              [6, 6], [7, 7], [8, 8]], dtype=float)
y = np.array([0, 0, 0, 0, 1, 1, 1, 1])  # 0 ve 1 sınıfları

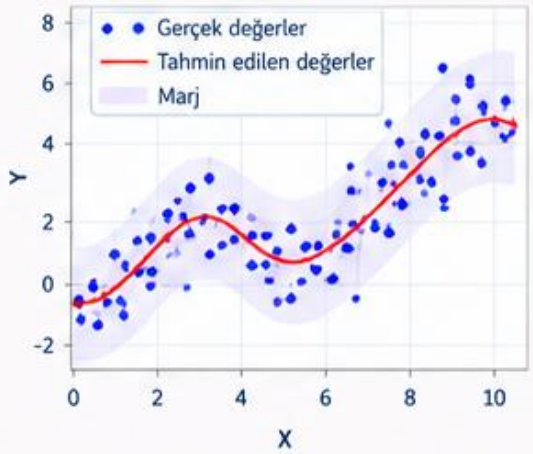
# Eğitim ve test verisi
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=42)

# SVM modeli
model = svm.SVC(kernel='linear')
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
print(f"Doğruluk: {accuracy_score(y_test, y_pred):.2f}")
```

Regresyon Örneği

SVM, regresyon problemlerinde de kullanılabilir.
SVR (Support Vector Regression) ile sürekli değerler tahmin edilir.



SVR, belirli bir hata sınırı içinde en iyi uyumu sağlayan modeli bulur.

Kernel Fonksiyonları

Kernel	Açıklama	Kullanım Alanı
Linear	Doğrusal ayrım.	Lineer ayrılabilir veriler
Polynomial	Polinom dönüşümü.	Daha karmaşık sınırlar
RBF (Gaussian)	Uzaklık tabanlı.	En yaygın kullanılan
Sigmoid	Sinir ağı benzeri.	Özel durumlar

Doğru kernel seçimi, modelin performansını önemli ölçüde etkiler.

Kullanım Alanları

- Metin ve belge sınıflandırma (spam tespiti, duygu analizi)
- Görüntü analizi (yüz tanıma, nesne tespiti)
- Biyoinformatik (hastalık sınıflandırma)
- Finans (kredi riski analizi)
- Pazarlama (müşteri segmentasyonu)

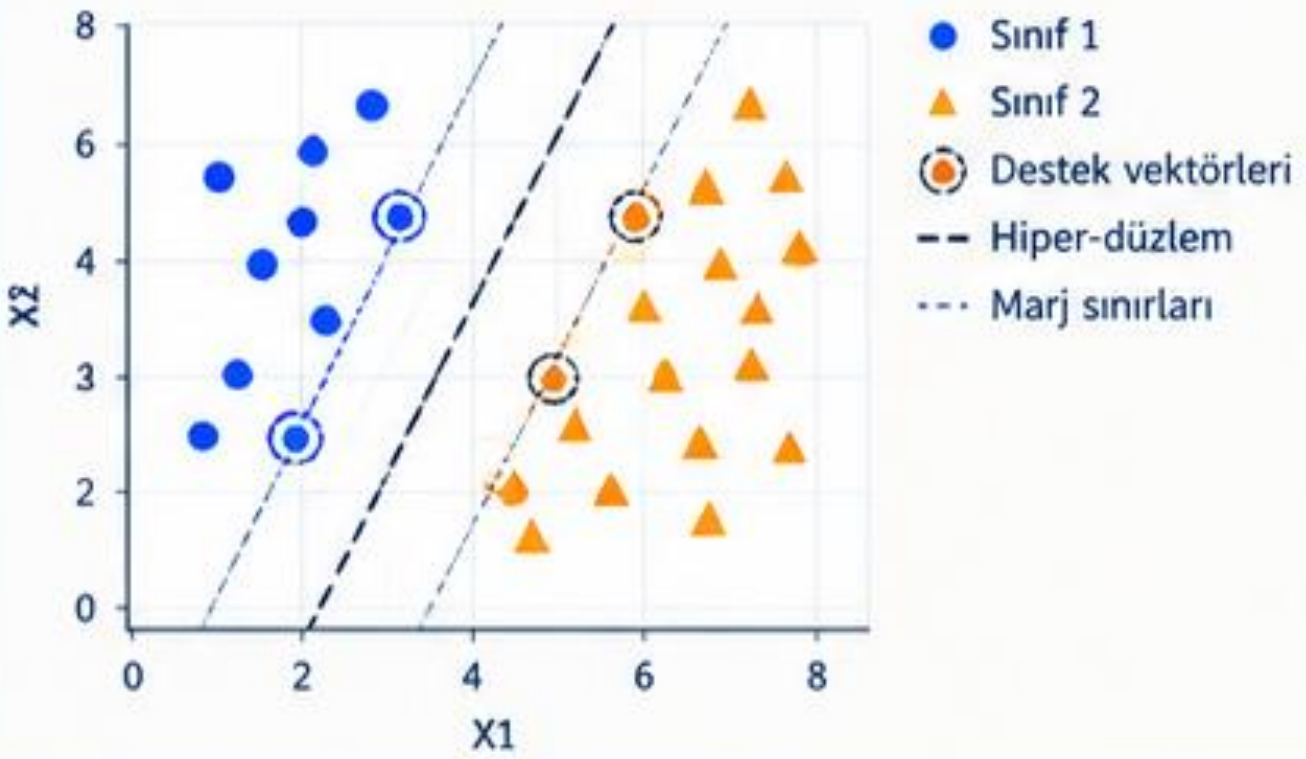
Avantajları

- Yüksek sınıflandırma doğruluğu.
- Küçük örneklemlerde etkilidir.
- Yüksek boyutlu veri setlerinde başarılıdır.
- Farklı kernel fonksiyonları ile esneklik sağlar.
- Overfitting riskini azaltır (özellikle doğru parametre seçildiğinde).

Dezavantajları

- Büyük veri setlerinde hesaplama maliyeti yüksektir.
- Kernel seçimi ve parametre tuning (C, γ) zordur.
- Sonuçların yorumlanması zordur (kara kutu gibi).
- Özellikle çok sayıda özellik olduğunda eğitim süresi uzayabilir.

Sınıflandırma Örneği (2 Boyutlu Veri)



SVM, sınıfları en iyi ayıran hiper-düzlemi bulur ve bu düzleme en yakın noktaları (destek vektörleri) kullanır.

Python Kodu (Sınıflandırma - Linear SVM)

```
import numpy as np
import matþıotlfib.pyplot as plt
from sklearn import svm
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Örnek veri seti
X = np.array([[1, 2], [2, 3], [3, 3], [4, 4], [5, 5],
              [6, 6], [7, 7], [8, 8]], dtype=float)
y = np.array([0, 0, 0, 0, 1, 1, 1, 1])  # 0 ve 1 sınıfları

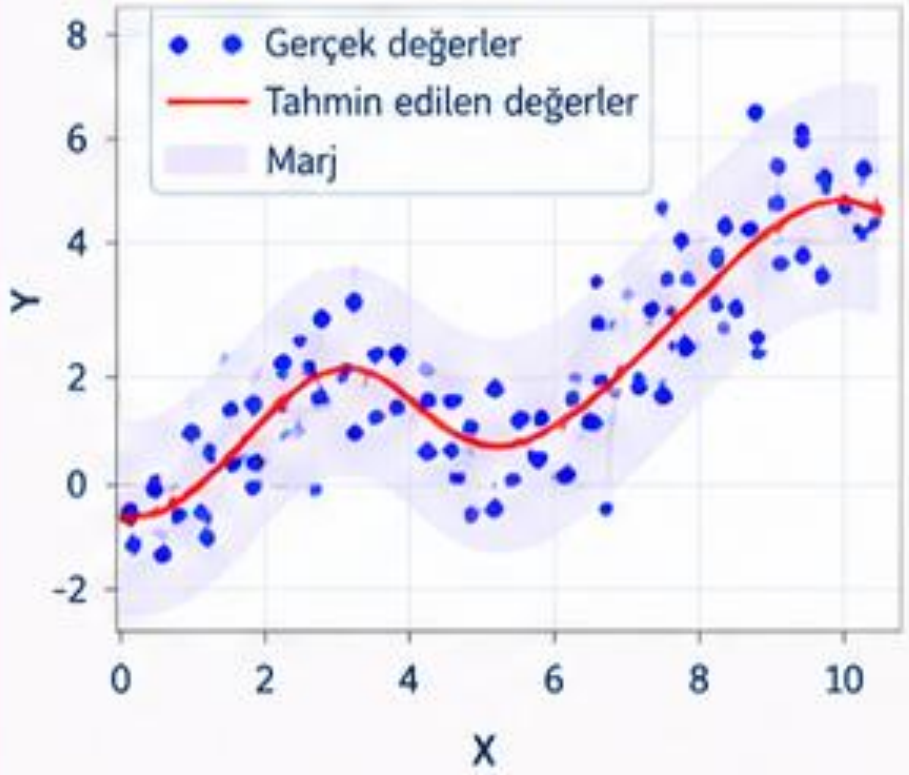
# Eğitim ve test verisi
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=42)

# SVM modeli
model = svm.SVC(kernel='linear')
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
print(f"Doğruluk: {accuracy_score(y_test, y_pred):.2f}")
```

Regresyon Örneği

SVM, regresyon problemlerinde de kullanılabilir.
SVR (Support Vector Regression) ile sürekli değerler tahmin edilir.



SVR, belirli bir hata sınırı içinde en iyi uyumu sağlayan modeli bulur.

Yapay Zeka ve Uygulamaları - Copyright © 2026

Kernel Fonksiyonları

Kernel	Açıklama	Kullanım Alanı
Linear	Doğrusal ayrım.	Lineer ayrılabilir veriler
Polynomial	Polinom dönüşümü.	Daha karmaşık sınırlar
RBF (Gaussian)	Uzaklık tabanlı.	En yaygın kullanılan
Sigmoid	Sinir ağı benzeri.	Özel durumlar

Doğru kernel seçimi, modelin performansını önemli ölçüde etkiler.

Kullanım Alanları

- Metin ve belge sınıflandırma (spam tespiti, duygu analizi)
- Görüntü analizi (yüz tanıma, nesne tespiti)
- Biyoinformatik (hastalık sınıflandırma)
- Finans (kredi riski analizi)
- Pazarlama (müşteri segmentasyonu)

Avantajları

- Yüksek sınıflandırma doğruluğu.
- Küçük örneklemlerde etkilidir.
- Yüksek boyutlu veri setlerinde başarılıdır.
- Farklı kernel fonksiyonları ile esneklik sağlar.
- Overfitting riskini azaltır (özellikle doğru parametre seçildiğinde).

Dezavantajları

- Büyük veri setlerinde hesaplama maliyeti yüksektir.
- Kernel seçimi ve parametre tuning (C, γ) zordur.
- Sonuçların yorumlanması zordur (kara kutu gibi).
- Özellikle çok sayıda özellik olduğunda eğitim süresi uzayabilir.

Makine Öğrenmesi Algoritmaları

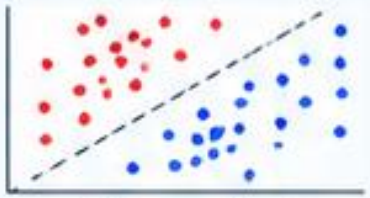
Verilerden öğrenen, tahmin yapan ve karar destek sağlayan algoritmalar

Denetimli Öğrenme (Süpervizyonlu) Algoritmaları

Etiketli verilerle (girdi + çıktı) çalışır. Sınıflandırma ve regresyon problemlerinde kullanılır.

Sınıflandırma Algoritmaları

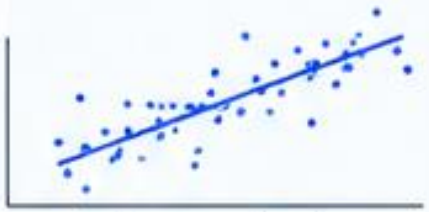
1. Doğrusal Regresyon (Linear Regression)
2. Lojistik Regresyon (Logistic Regression)
3. Naive Bayes
4. K-Nearest Neighbors (KNN)
5. Destek Vektör Makineleri (SVM)
6. Karar Ağaçları (Decision Tree)
7. Rastgele Orman (Random Forest)
8. Gradyan Boosting (GBM, XGBoost, LightGBM, CatBoost)
9. AdaBoost
10. Çok Katmanlı Algılayıcı (MLP)



Sınıflandırma Örneği

Regresyon Algoritmaları

1. Doğrusal Regresyon
2. Ridge Regresyon
3. Lasso Regresyon
4. Elastic Net
5. SVR (Support Vector Regression)
6. Karar Ağaçları Regresyonu
7. Rastgele Orman Regresyonu
8. Gradient Boosting Regresyonu
9. XGBoost / LightGBM
10. CatBoost



Regresyon Örneği

Denetimsiz Öğrenme (Unsupervizıon) Algoritmaları

Etiketsiz verilerle çalışır. Veri içindeki gizli yapıları, kümeleri ve örüntüleri bulur.

Kümeleme (Clustering)

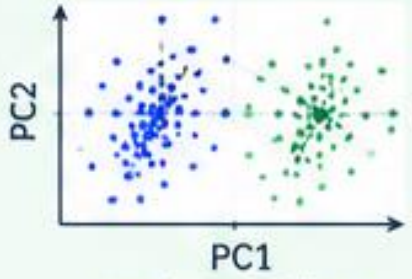
1. K-Means
2. DBSCAN
3. Hierarchical (Hiyerarşik) Kümeleme
4. Gaussian Mixture Models (GMM)
5. Fuzzy C-Means
6. Spectral Clustering
7. Mean Shift
8. Agglomerative Clustering



Kümeleme Örneği

Boyut İndirgeme (Dimensionality Reduction)

1. PCA (Temel Bileşenler Analizi)
2. t-SNE
3. UMAP
4. LDA (Lineer Diskriminant Analizi)
5. ICA (Bağımsız Bileşen Analizi)
6. MDS (Çok Boyutlu Ölçekleme)
7. Autoencoder (Derin Öğrenmede)

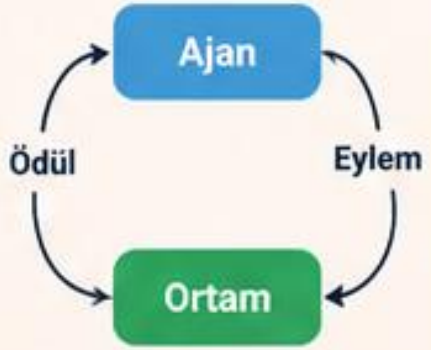


Boyut İndirgeme Örneği

Pekiştirmeli Öğrenme (Reinforcement Learning)

Ajanın çevresiyle etkileşim kurarak ödül (reward) maksimize etmeyi öğrenir.

1. Q-Learning
2. SARSA
3. Deep Q-Network (DQN)
4. Policy Gradient (REINFORCE)
5. Actor-Critic
6. A2C (Advantage Actor-Critic)
7. PPO (Proximal Policy Optimization)
8. DDPG (Deep Deterministic Policy Gradient)
9. TD3 (Twin Delayed DDPG)
10. SAC (Soft Actor-Critic)

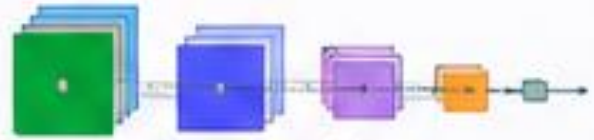


Pekiştirmeli Öğrenme Döngüsü

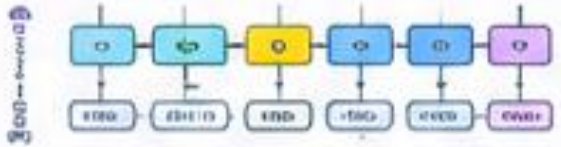
Derin Öğrenme (Deep Learning) Algoritmaları

Büyük veri ve çok katmanlı sinir ağları ile karmaşık problemleri çözer.

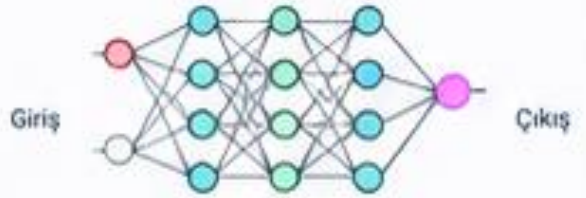
1. Çok Katmanlı Perceptron (MLP)
2. Evrişimli Sinir Ağları (CNN)
3. Reküren Sinir Ağları (RNN)
4. LSTM (Uzun Kısa Süreli Bellek)
5. GRU (Gated Recurrent Unit)
6. Transformer
7. BERT (Doğal Dil İşleme)
8. GPT (Üretken Yapay Zekâ)
9. GAN (Generative Adversarial Network)
10. Autoencoder
11. U-Net (Görüntü İşleme)
12. YOLO (Nesne Tespiti)



CNN – Görüntü İşleme



Transformer – Doğal Dil İşleme



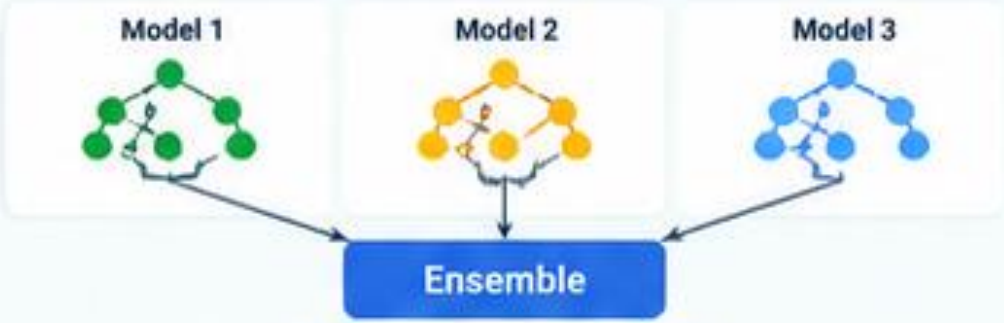
Yapay Sinir Ağı

Yapay Zeka ve Uygulamaları - Copyright © 2026

Ensemble (Topluluk) Algoritmaları

Birden fazla algoritmanın birlikte çalışmasıyla daha iyi performans sağlar.

1. Bagging (Bootstrap Aggregating)
2. Boosting (AdaBoost, Gradient Boosting)
3. Random Forest
4. Extra Trees
5. Stacking (Katmanlama)
6. Blending
7. XGBoost
8. LightGBM
9. CatBoost



Topluluk (Ensemble) Modeli

Diğer Önemli Algoritmalar

Belirli problemlere yönelik özel algoritmalardır.

1. Apriori (Birliktelik Kuralı Madenciliği)
2. FP-Growth
3. PageRank
4. K-Means++ (Gelişmiş başlatma)
5. Mean Shift
6. Spectral Clustering
7. Isomap
8. Gaussian Process (GPR)
9. Markov Chain (Markov Modelleri)
10. Hidden Markov Model (HMM)
11. CRF (Conditional Random Field)
12. Word2Vec (Kelime Gömme)



Birliktelik Kuralı (Apriori)



PageRank



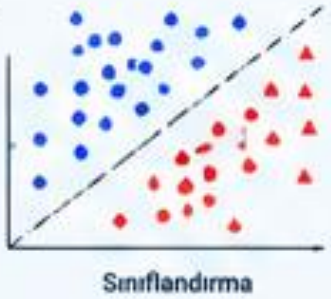
HMM (Gizli Markov Modeli)

Yapay Zeka ve Uygulamaları - Copyright © 2026

Regresyon ve Sınıflandırma Örnek Karşılaştırması

Sınıflandırma

- Lojistik Regresyon
- SVM
- Karar Ağacı
- Random Forest
- XGBoost
- Naive Bayes



Sınıflandırma

Regresyon

- Lineer Regresyon
- Ridge / Lasso
- SVR
- Random Forest Reg.
- Gradient Boosting
- XGBoost

● Sınıf 1
▲ Sınıf 2



Regresyon

Algoritmaların Kullanım Alanları

Sağlık	Finans	Eğitim	Pazarlama
Görüntü İşleme	NLP	Otomotiv	Tarım
Biyoinformatik	Enerji	Üretim	Siber Güvenlik

Doğru Algoritma Nasıl Seçilir?

- ✅ Problemin türü (sınıflandırma, regresyon, kümeleme vb.)
- ✅ Veri setinin büyüklüğü ve yapısı
- ✅ Etiketli / etiketsiz veri durumu
- ✅ Performans metrikleri (Doğruluk, F1, RMSE vb.)
- ✅ Hesaplama maliyeti ve hız
- ✅ Yorumlanabilirlik (açıklanabilirlik)
- ✅ Uygulama alanının gereksinimleri

Doğru algoritma + Doğru veri + Doğru değerlendirme = Başarılı bir makine öğrenmesi projesi