

### MAKİNE ÖĞRENMESİ

# MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ

**Hazırlayanlar:**

Zehra KARAGÜL

Zeynep YILDIZ

Sedanur YİĞİT

Sıla TOPUZ

Rabia ÖRS

İstanbul, 2024

# ÖNSÖZ

Makine öğrenmesi, son yıllarda teknoloji dünyasında büyük bir devrim yaratmış ve pek çok sektörde dönüşümü hızlandıran önemli bir alan olmuştur. İnsan müdahalesi olmadan verilerden öğrenme ve tahmin yapabilme yeteneği, makine öğrenmesini yalnızca bilimsel araştırmalardan iş dünyasına kadar geniş bir yelpazede etkili bir araç haline getirmiştir. Bu makale, makine öğrenmesinin temellerinden günümüzdeki uygulama alanlarına kadar geniş bir perspektifte ele alınarak, bu alanın tarihçesi, mevcut kullanımı, karşılaşılan zorluklar ve geleceği hakkında derinlemesine bilgi sunmayı amaçlamaktadır.

Makine öğrenmesinin geçmişi, 1950’ler gibi erken bir döneme dayansa da günümüzde büyük veri, yapay zekâ ve derin öğrenme alanlarındaki gelişmelerle birlikte bu teknoloji hızla evrilmiştir. Bu makalede, makine öğrenmesinin tarihsel süreci, temel kavramlar ve türler, kullanılan algoritmalar ve modeller ayrıntılı bir şekilde ele alınmaktadır. Ayrıca, makine öğrenmesinin sağlık, finans, otomotiv gibi farklı sektörlerde nasıl kullanıldığına dair örnekler sunularak, bu teknolojinin günlük yaşantımıza nasıl etki ettiği vurgulanmaktadır.

Makine öğrenmesinin geleceği, yeni teknolojiler ve algoritmalarla şekillenmeye devam etmektedir. Bu makalede, güncel gelişmeler ve gelecekteki potansiyel trendler de tartışılmakta olup, okuyuculara makine öğrenmesinin gelecekteki evrimini anlamada bir rehber sunulmaktadır. Son olarak, makine öğrenmesinin etik ve güvenlik gibi önemli meselelerine de değinilerek, bu teknolojinin daha sorumlu bir şekilde nasıl kullanılabileceği üzerine düşünceler paylaşılmaktadır.

Bu çalışma, makine öğrenmesi alanındaki temel bilgileri hem yeni başlayanlar hem de konuyu derinlemesine incelemek isteyenler için kapsamlı bir kaynak olmayı hedeflemektedir.

**İçindekiler**

ÖNSÖZ………………………………………………………………………………………………………………….i

[ŞEKİLLER LİSTESİ ix](#_Toc185677429)

[TABLOLAR LİSTESİ xii](#_Toc185677430)

[MAKİNE ÖĞRENMESİ 13](#_Toc185677431)

[MAKİNE ÖĞRENMESİ NEDİR? 13](#_Toc185677432)

[MAKİNE ÖĞRENMESİNİN TARİHÇESİ 13](#_Toc185677433)

[1950’ler: Temellerin Atılması 13](#_Toc185677434)

[1960’lar: İlk Algoritmalar ve Sinir Ağları 14](#_Toc185677435)

[1970’ler: Kural Tabanlı Sistemler 14](#_Toc185677436)

[1980’ler: Yeni Yaklaşımlar 14](#_Toc185677437)

[1990’lar: Veri Madenciliği ve İstatistiksel Öğrenme 14](#_Toc185677438)

[2000’ler: Derin Öğrenme ve Büyük Veri 15](#_Toc185677439)

[Günümüz 15](#_Toc185677440)

[MAKİNE ÖĞRENMESİ HANGİ ALANLARDA KULLANILIR? 15](#_Toc185677441)

[Sağlık 15](#_Toc185677442)

[Finans 16](#_Toc185677443)

[Otomotiv 16](#_Toc185677444)

[Perakende ve E-Ticaret 16](#_Toc185677445)

[Lojistik ve Ulaşım 17](#_Toc185677446)

[Eğitim 17](#_Toc185677447)

[Hukuk 17](#_Toc185677448)

[MAKİNE ÖĞRENMESİNDE TEMEL KAVRAMLAR 18](#_Toc185677449)

[Veri (Data) nedir? 18](#_Toc185677450)

[Veri Ön İşleme (Data Preprocessing) nedir? 18](#_Toc185677451)

[Özellik Mühendisliği (Feature Engineering) nedir? 19](#_Toc185677452)

[Model ve algoritma nedir? 20](#_Toc185677453)

[Eğitim, doğrulama ve test nedir? 20](#_Toc185677454)

[NASIL ÇALIŞIR? 23](#_Toc185677455)

[1. Karar Süreci: 23](#_Toc185677456)

[2. Hata İşlevi: 23](#_Toc185677457)

[3. Model Optimizasyon Süreci 23](#_Toc185677458)

[MAKİNE ÖĞRENMESİ TÜRLERİ NELERDİR? 24](#_Toc185677459)

[Denetimli Öğrenme (Supervised Learning) *Nedir?* 24](#_Toc185677460)

[ *Sınıflandırma:* 24](#_Toc185677461)

[ *Regresyon:* 24](#_Toc185677462)

[1. *Karar Ağaçları ve Rastgele Ormanlar:* 24](#_Toc185677463)

[2. *Destek Vektör Makineleri (SVM):* 24](#_Toc185677464)

[3. *Sinir Ağları:* 24](#_Toc185677465)

[4. *Doğrusal ve Lojistik Regresyon*: 24](#_Toc185677466)

[ *Veri Kalitesi:* 25](#_Toc185677467)

[ *Aşırı Uyum (Overfitting):* 25](#_Toc185677468)

[ *Hesaplama Gücü:* 25](#_Toc185677469)

[Denetimsiz Öğrenme (Unsupervised Learning) Nedir? 25](#_Toc185677470)

[2. *Hiyerarşik Kümeleme Analizi (Hierarchical Cluster Analysis)* 26](#_Toc185677471)

[3. Temel Bileşen Analizi (Principal Component Analysis, PCA) 27](#_Toc185677472)

[Pekiştirmeli Öğrenme (Reinforcement Learning — RL) Nedir? 27](#_Toc185677473)

[*Ajan (Agent):* 27](#_Toc185677474)

[*Çevre (Environment):* 27](#_Toc185677475)

[*Durum (State):* 27](#_Toc185677476)

[*Aksiyon (Action):* 27](#_Toc185677477)

[*Ödül (Reward):* 27](#_Toc185677478)

[1*. Robotik* 28](#_Toc185677479)

[2*. Oyunlar* 29](#_Toc185677480)

[3*. Otonom Araçlar* 29](#_Toc185677481)

[4*. Finans* 29](#_Toc185677482)

[5*.Sağlık* 29](#_Toc185677483)

[6*.Doğal Dil İşleme (NLP)* 29](#_Toc185677484)

[7*. Enerji Yönetimi* 29](#_Toc185677485)

[8*. Reklamcılık ve Öneri Sistemleri* 29](#_Toc185677486)

[MAKİNE ÖĞRENMESİNDE ALGORİTMALAR VE MODELLER 30](#_Toc185677487)

[GİRİŞ 30](#_Toc185677488)

[Makine Öğrenmesi Algoritması Seçerken Göz Önünde Bulundurulması Gereken Faktörler 32](#_Toc185677489)

[1. Veri Türü 32](#_Toc185677490)

[2. Problemin Karmaşıklığı 32](#_Toc185677491)

[3. Hesaplama Kaynakları 32](#_Toc185677492)

[4. Yorumlanabilirlik ve Doğruluk 33](#_Toc185677493)

[Makine Öğrenimi Algoritmaları Nasıl Çalışır? 33](#_Toc185677494)

[Karar Süreci: 33](#_Toc185677495)

[Model Optimizasyonu Süreci: 33](#_Toc185677496)

[SINIFLANDIRMA ALGORİTMALARI 34](#_Toc185677497)

[Destek Vektör Makineleri (SVM - Support Vector Machines) 34](#_Toc185677498)

[Karar Ağaçları (Decision Trees) 36](#_Toc185677499)

[Sinir Ağları (Neural Networks) 37](#_Toc185677500)

[K-En Yakın Komşular (k-NN) 40](#_Toc185677501)

[Lojistik Regresyon 42](#_Toc185677502)

[Rastgele Orman (Random Forest) 44](#_Toc185677503)

[Gradient Boosting ve AdaBoosting Algoritmaları 45](#_Toc185677504)

[Naive Bayes Algoritması 48](#_Toc185677505)

[REGRESYON ALGORİTMALARI 49](#_Toc185677506)

[Doğrusal Regresyon 50](#_Toc185677507)

[Destek Vektör Regresyonu (SVR) 51](#_Toc185677508)

[Ridge Regresyonu 52](#_Toc185677509)

[Lasso Regresyonu 53](#_Toc185677510)

[KÜMELEME ALGORİTMALARI 54](#_Toc185677511)

[Hiyerarşik Kümeleme 54](#_Toc185677512)

[DBSCAN (Gürültüye Sahip Uygulamaların Yoğunluk Tabanlı Mekansal Kümelenmesi) 56](#_Toc185677513)

[Gauss Karışım Modelleri (GMM) 57](#_Toc185677514)

[K-Means Algoritması 58](#_Toc185677515)

[BOYUT AZALTMA ALGORİTMALARI 59](#_Toc185677516)

[Temel Bileşen Analizi (PCA) 59](#_Toc185677517)

[T-Dağıtılmış Stokastik Komşu Gömme (T-SNE) 62](#_Toc185677518)

[Doğrusal Diskriminant Analizi (LDA) 63](#_Toc185677519)

[Bağımsız Bileşen Analizi (ICA) 65](#_Toc185677520)

[UMAP (Uniform Manifold Approximation and Projection) 67](#_Toc185677521)

[ANOMALİ TESPİTİ 68](#_Toc185677522)

[Yerel Aykırı Değer Faktörü (LOF - Local Outlier Factor) 69](#_Toc185677523)

[İzolasyon Ormanı (Isolation Forest) 70](#_Toc185677524)

[APRİORİ ALGORİTMASI 72](#_Toc185677525)

[TORBALAMA (BAGGING) 73](#_Toc185677526)

[ECLAT ALGORİTMASI 75](#_Toc185677527)

[Derin öğrenme nedir ? 76](#_Toc185677528)

[Derin öğrenme nasıl çalışır? 76](#_Toc185677529)

[Yapay Sinir Ağı Nedir? 77](#_Toc185677530)

[Yapay Sinir Ağının Yapısı 77](#_Toc185677531)

[Evrişimsel Sinir Ağları (Convolutional Neural Network-CNN) 78](#_Toc185677532)

[CNN Katmanları: 78](#_Toc185677533)

[CNN'nin Avantajları: 78](#_Toc185677534)

[Kullanım Alanları: 79](#_Toc185677535)

[CNN Nasıl Çalışır ? 79](#_Toc185677536)

[Peki bu süreç neden önemli? 80](#_Toc185677537)

[Tekrarlayan Sinir Ağları (Recurrent Neural Network-RNN) 82](#_Toc185677538)

[RNN'nin Temel Özellikleri 83](#_Toc185677539)

[RNN’nin Yapısı 83](#_Toc185677540)

[RNN'nin Avantajları 84](#_Toc185677541)

[Tekrarlayan Sinir Ağı Uygulamaları (RNN Applications) 85](#_Toc185677542)

[Derin öğrenme ve sinir ağları arasındaki fark nedir? 85](#_Toc185677543)

[Transfer Öğrenme Nedir? 86](#_Toc185677544)

[Öğrenme aktarımı sürecinde 3 soru sorulur: 88](#_Toc185677545)

[Transfer Öğrenmenin Aşamaları 89](#_Toc185677546)

[Transfer Öğrenmenin Avantajları 89](#_Toc185677547)

[Transfer Öğrenme Yöntemleri 90](#_Toc185677548)

[Transfer Öğrenmenin Kullanım Alanları 91](#_Toc185677549)

[Makine Öğrenimi ve Derin Öğrenme Arasındaki Fark Nedir? 92](#_Toc185677550)

[Veri Seti Nedir? 92](#_Toc185677551)

[Veri Seti Türleri 93](#_Toc185677552)

[*1. Eğitim Verisi (Training Dataset)* 93](#_Toc185677553)

[*2. Doğrulama Verisi (Validation Dataset)* 93](#_Toc185677554)

[*3. Test Verisi (Test Dataset)* 93](#_Toc185677555)

[*4. Yapılandırılmış ve Yapılandırılmamış Veri* 93](#_Toc185677556)

[*5. Zaman Serisi Verisi* 93](#_Toc185677557)

[*6. Kategorik ve Sayısal Veri* 94](#_Toc185677558)

[7. Büyük Veri Setleri (Big Data) 94](#_Toc185677559)

[Veri Kümesinin Temel Analitik Özellikleri 94](#_Toc185677560)

[*1. Veri Merkezi (Central Tendency)* 94](#_Toc185677561)

[*2. Verilerin Çarpıklığı (Skewness)* 94](#_Toc185677562)

[*3. Veri Yayılımı (Dispersion)* 95](#_Toc185677563)

[*4. Aykırı Değerler (Outliers)* 95](#_Toc185677564)

[*5. Korelasyon (Correlation)* 95](#_Toc185677565)

[Veri Kümesinin Yapısal ve Format Özellikleri 95](#_Toc185677566)

[*1. Sayısal Özellikler (Numerical Features)* 95](#_Toc185677567)

[*2. Kategorik Özellikler (Categorical Features)* 95](#_Toc185677568)

[*3. Meta Veriler (Metadata)* 96](#_Toc185677569)

[*4. Veri Boyutu (Data Size)* 96](#_Toc185677570)

[*5.Veri Biçimlendirilmesi (Data Formatting)* 96](#_Toc185677571)

[*6. Hedef Değişken (Target Variable)* 96](#_Toc185677572)

[Veri Setlerinde Kullanılan Yöntemler 97](#_Toc185677573)

[*1. Veri Kümelerini Yükleme ve Okuma:* 97](#_Toc185677574)

[*2. Keşifsel Veri Analizi:* 97](#_Toc185677575)

[*3. Veri Ön İşleme:* 97](#_Toc185677576)

[*4. Veri Manipülasyonu:* 98](#_Toc185677577)

[*5. Görselleştirmeyi Verin:* 98](#_Toc185677578)

[*6. Veri İndeksleme, Veri Alt Kümeleri:* 98](#_Toc185677579)

[*7. Dış Veri Aktarımı:* 98](#_Toc185677580)

[Veri Setlerinin Kullanım Alanları 98](#_Toc185677581)

[*1. Makine Öğrenmesi ve Yapay Zekâ* 98](#_Toc185677582)

[*2. Sağlık Bilimleri* 99](#_Toc185677583)

[*3. Çevre Bilimleri* 99](#_Toc185677584)

[*4. Ekonomik ve Finansal Analizler* 99](#_Toc185677585)

[*5. Sosyal Bilimler* 99](#_Toc185677586)

[Değerlendirme Metrikleri 100](#_Toc185677587)

[*1. Doğruluk (Accuracy)* 100](#_Toc185677588)

[*2. Hata Oranı (Error Rate)* 100](#_Toc185677589)

[*3. F1 Skoru (F1 Score)* 101](#_Toc185677590)

[MAKİNE ÖĞRENMESİNDE KARŞILAŞILAN SORUNLAR 101](#_Toc185677591)

[1. AŞIRI UYUM (OVERFIT) 101](#_Toc185677592)

[2. EKSİK UYUM (UNDERFIT) 102](#_Toc185677593)

[3. VERİ DENGESİZLİĞİ 102](#_Toc185677594)

[4. HATALI ÖZELLİK MÜHENDİSLİĞİ 103](#_Toc185677595)

[5. MODEL SEÇİMİ VE HİPERPARAMETRE OPTİMİZASYONU 104](#_Toc185677596)

[MAKİNE ÖĞRENMESİNİN DİĞER ALANLARLA FARKLARI VE İLİŞKİLERİ 105](#_Toc185677597)

[*Yapay Zekâ* *(AI) ve Makine Öğrenmesi* 105](#_Toc185677598)

[*Veri Bilimi ve Makine Öğrenmesi* 105](#_Toc185677599)

[*Derin Öğrenme ve Makine Öğrenmesi* 106](#_Toc185677600)

[*İstatistik ve Makine Öğrenmesi* 106](#_Toc185677601)

[*Bilgisayarla Görüntü İşleme ve Makine Öğrenmesi* 106](#_Toc185677602)

[*Doğal Dil İşleme (NLP) ve Makine Öğrenmesi* 106](#_Toc185677603)

[*Big Data ve Makine Öğrenmesi* 106](#_Toc185677604)

[MAKİNE ÖĞRENMESİNDEKİ RİSKLER 107](#_Toc185677605)

[*Önyargı ve Ayrımcılık:* 107](#_Toc185677606)

[*Veri Gizliliği ve Güvenliği:* 107](#_Toc185677607)

[*Derin Sahtecilik (Deepfake):* 107](#_Toc185677608)

[*Adversarial Saldırılar:* 108](#_Toc185677609)

[*Şeffaflık Eksikliği:* 108](#_Toc185677610)

[*İş Kaybı ve Sosyoekonomik Etkiler:* 108](#_Toc185677611)

[*Hatalı Kararlar:* 109](#_Toc185677612)

[*Kötüye Kullanım Potansiyeli:* 109](#_Toc185677613)

[*Regülasyon Eksikliği:* 109](#_Toc185677614)

[*İnsan Yetkinliklerinde Azalma:* 109](#_Toc185677615)

[RİSKLERE KARŞI NASIL ÖNLEMLER ALINABİLİR? 109](#_Toc185677616)

[MAKİNE ÖĞRENMESİNDE ETİK 110](#_Toc185677617)

[*Önyargı ve Ayrımcılık:* 110](#_Toc185677618)

[*Şeffaflık ve Hesap Verebilirlik:* 110](#_Toc185677619)

[*Gizlilik ve Veri Koruma:* 110](#_Toc185677620)

[*Adalet ve Eşitlik:* 110](#_Toc185677621)

[GÜNCEL GELİŞMELER VE TRENDLER 111](#_Toc185677622)

[*Otomatik Makine Öğrenmesi (AutoML)* 111](#_Toc185677623)

[*Açıklanabilir Yapay Zekâ (XAI)* 111](#_Toc185677624)

[*Düşük Kod ve Kodsuz Platformlar* 111](#_Toc185677625)

[*Sorumlu Makine Öğrenmesi ve Etik AI* 112](#_Toc185677626)

[*Kuantum Makine Öğrenmesi* 112](#_Toc185677627)

[Yapay Zekâ ve Nesnelerin İnterneti (IoT) Entegrasyonu 112](#_Toc185677628)

[MAKİNE ÖĞRENMESİNİN GELECEKTEKİ POTANSİYELİ 113](#_Toc185677629)

[Kaynakça 114](#_Toc185677630)

# ŞEKİLLER LİSTESİ

[Şekil 1: Turing Testi 13](#_Toc185422050)

[Şekil 2 18](#_Toc185422051)

[Şekil 3 19](#_Toc185422052)

[Şekil 4 21](#_Toc185422053)

[Şekil 5 24](#_Toc185422054)

[Şekil 6: Denetimli Makine Öğrenmesi 25](#_Toc185422055)

[Şekil 7: Makine öğrenmesi türleri 30](#_Toc185422056)

[Şekil 8: Makine Öğrenmesi Türleri ve Algoritmalar 31](#_Toc185422057)

[Şekil 9: Makine Öğrenmesi Türleri ve Algoritmaları Gösterimi 31](#_Toc185422058)

[Şekil 10: Makine Öğrenmesi Nasıl Çalışır? 33](#_Toc185422059)

[Şekil 11: Sınıflandırma Algoritması Gösterimi 34](#_Toc185422060)

[Şekil 12: Karar Ağaçları Gösterimi 36](#_Toc185422061)

[Şekil 13: Sinir Ağları Gösterimi 38](#_Toc185422062)

[Şekil 14: K-NN Algoritması Örnek Gösterimi 40](#_Toc185422063)

[Şekil 15: Rastgele Orman Algoritması Gösterimi 44](#_Toc185422064)

[Şekil 16: Naive Bayes Algoritması Gösterimi 48](#_Toc185422065)

[Şekil 17: Regresyon Algoritmaları 49](#_Toc185422066)

[Şekil 18: Ridge Regresyon 52](#_Toc185422067)

[Şekil 19: Yapay Sinir Ağları Katmanları 77](#_Toc185422068)

[Şekil 20: CNN Örneği 79](#_Toc185422069)

[Şekil 21: CNN Örneği 80](#_Toc185422070)

[Şekil 22: CNN Örneği 81](#_Toc185422071)

[Şekil 23 81](#_Toc185422072)

[Şekil 24: Düzleştirme Katmanı Örneği 82](#_Toc185422073)

[Şekil 25 83](#_Toc185422074)[Şekil 1: Turing Testi 13](#_Toc185422050)

[Şekil 2 18](#_Toc185422051)

[Şekil 3 19](#_Toc185422052)

[Şekil 4 21](#_Toc185422053)

[Şekil 5 24](#_Toc185422054)

[Şekil 6: Denetimli Makine Öğrenmesi 25](#_Toc185422055)

[Şekil 7: Makine öğrenmesi türleri 30](#_Toc185422056)

[Şekil 8: Makine Öğrenmesi Türleri ve Algoritmalar 31](#_Toc185422057)

[Şekil 9: Makine Öğrenmesi Türleri ve Algoritmaları Gösterimi 31](#_Toc185422058)

[Şekil 10: Makine Öğrenmesi Nasıl Çalışır? 33](#_Toc185422059)

[Şekil 11: Sınıflandırma Algoritması Gösterimi 34](#_Toc185422060)

[Şekil 12: Karar Ağaçları Gösterimi 36](#_Toc185422061)

[Şekil 13: Sinir Ağları Gösterimi 38](#_Toc185422062)

[Şekil 14: K-NN Algoritması Örnek Gösterimi 40](#_Toc185422063)

[Şekil 15: Rastgele Orman Algoritması Gösterimi 44](#_Toc185422064)

[Şekil 16: Naive Bayes Algoritması Gösterimi 48](#_Toc185422065)

[Şekil 17: Regresyon Algoritmaları 49](#_Toc185422066)

[Şekil 18: Ridge Regresyon 52](#_Toc185422067)

[Şekil 19: Yapay Sinir Ağları Katmanları 77](#_Toc185422068)

[Şekil 20: CNN Örneği 79](#_Toc185422069)

[Şekil 21: CNN Örneği 80](#_Toc185422070)

[Şekil 22: CNN Örneği 81](#_Toc185422071)

[Şekil 23 81](#_Toc185422072)

[Şekil 24: Düzleştirme Katmanı Örneği 82](#_Toc185422073)

[Şekil 25 83](#_Toc185422074)

[Şekil 26: Transfer Öğrenme 87](#_Toc185422075)

[Şekil 27: Transfer Öğrenme Süreci 88](#_Toc185422076)

[Şekil 28: Süreç 89](#_Toc185422077)

[Şekil 29: Makine Öğrenmesi ve Derin Öğrenme Arasındaki Fark 92](#_Toc185422078)

[Şekil 30 97](#_Toc185422079)

[Şekil 31: Yapay Zekâ - Makine Öğrenmesi - Derin Öğrenme İlişkisi 105](#_Toc185422080)

[Şekil 32: Derin Sahtecilik Örneği 107](#_Toc185422081)

[Şekil 33: Adversarial Saldırı Örneği 108](#_Toc185422082)

# TABLOLAR LİSTESİ

[Tablo 1: Model türleri ve kullanım alanları 20](#_Toc185422083)

[Tablo 2: Model değerlendirmesi ve testi 22](#_Toc185422084)

# MAKİNE **ÖĞRENMESİ**

## MAKİNE ÖĞRENMESİ NEDİR?

Makine öğrenimi bilgisayarların geçmiş verileri analiz ederek öğrenmesini ve bu öğrendiklerini kullanarak kararlar almasını ya da tahminler yapmasını sağlayan bir yapay zekâ dalıdır. İnsan zekâsını taklit etmeyi amaçlayan algoritmalar, belirli bir problemin uygun verilerle modellenmesini sağlar. Bu teknoloji bilgisayar sistemlerinin büyük miktarda veriyi işleyip örüntüleri tanımlayarak insanlar tarafından açıkça kodlanmadan deneyimden öğrenmesine olanak tanır. Örneğin bir tıbbi uygulama milyonlarca tarama görüntüsü ve teşhisi analiz ederek röntgen görüntülerinden kanseri teşhis etmeyi öğrenebilir.

Makine öğrenimi temel olarak veri ve algoritmalar üzerinde çalışır. Algoritmalar verilerdeki desenleri keşfeder ve bu bilgiyi yeni durumlara uygulayarak sonuçları daha doğru bir şekilde tahmin eder. Örneğin Spotify veya Netflix gibi platformlar, geçmişte dinlediğiniz şarkıları veya izlediğiniz filmleri analiz ederek size sevebileceğiniz içerikleri önerir.

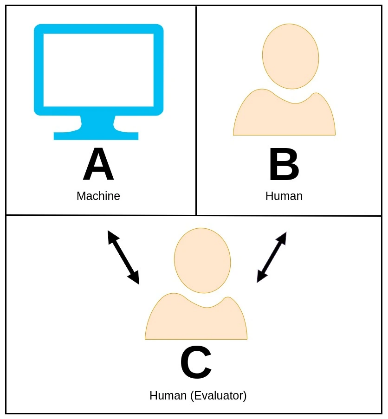
Makine Öğreniminin Öne Çıkan Özellikleri:

* Geçmiş verilerden öğrenme yeteneği
* Kodlama yerine, örüntüler ve ilişkiler üzerine kurulu bir öğrenme sistemi
* Algoritmalar sayesinde veri analizi ve tahmin doğruluğunun artırılması

## MAKİNE ÖĞRENMESİNİN TARİHÇESİ

Makine öğrenmesinin tarihçesi yapay zekâ araştırmalarının temel taşlarından biri olarak birçok önemli gelişme içermektedir:

### 1950’ler: Temellerin Atılması



Alan Turing, 1950 yılında yayımladığı “Computing Machinery and Intelligence” makalesinde makinelerin insan gibi düşünebilme yeteneğini ölçmek için Turing Testini ortaya koydu. Bu test, bir makinenin verdiği yanıtların bir insanın yanıtlarından ayırt edilememesi durumunda başarılı sayılmasını temel alır. Turing'in bu fikri yapay zekâ ve makine öğrenmesi araştırmalarına yön veren bir kilometre taşıdır.

1952’de Arthur Samuel, IBM’de çalışırken dama oynayan bir makine öğrenme programı geliştirdi. Bu program, algoritmaların geçmiş deneyimlerden öğrenerek performanslarını geliştirdiğini gösteren ilk örneklerden biridir. Samuel'in çalışması "öğrenen makineler" kavramının temellerini atmış ve bu alandaki ilk büyük adımlardan birini oluşturmuştur.

**Şekil 1: Turing Testi**

### 1960’lar: İlk Algoritmalar ve Sinir Ağları

1967’de En Yakın Komşu (Nearest Neighbor) algoritması geliştirildi. Bu yöntem temel sınıflandırma problemlerini çözmek için kullanılan ilk algoritmalardan biridir. En Yakın Komşu (Nearest Neighbor) algoritması, bir veri noktasının sınıfını ona en yakın olan diğer veri noktalarının sınıflarına bakarak tahmin eder. Örneğin bir e-posta spam mı değil mi diye sorulduğunda algoritma daha önce "spam" ya da "spam olmayan" olarak etiketlenmiş e-postaların en yakın komşularını bulur ve çoğunluk sınıfına göre karar verir. Bu yöntem özellikle veri setindeki benzerliklere dayanarak basit ve etkili bir sınıflandırma yapar. Ayrıca yapay sinir ağları konsepti bu dönemde ilk kez öne çıktı. Ancak o dönemin işlem gücündeki sınırlılıklar nedeniyle bu ağlar geniş bir uygulama alanı bulamamıştı.

### 1970’ler: Kural Tabanlı Sistemler

1970'ler kural tabanlı yapay zekâ sistemlerinin öne çıktığı bir dönemdi. Kural tabanlı sistemler bir problemin çözülmesi için kuralların belirlendiği ve bu kuralların sırayla uygulandığı yapay zekâ sistemleridir. Örneğin bir trafik simülasyonunda kural tabanlı bir sistem trafik ışıkları ve yol işaretlerine göre hareket eder. Eğer trafik ışığı kırmızıysa sistem durma komutunu verir; yeşilse ilerlemeye devam eder. Ayrıca yol işaretleri gibi ek kurallar da uygulanır, örneğin "dur" işareti görünüyorsa sistem aracı durdurur. Bu sistemler belirli kurallar çerçevesinde işlem yaparak insan kararlarını taklit eder ve bir durumu doğru şekilde yönetir.

Bu dönemde makine öğrenmesi araştırmalarında ilerleme yavaşladı. Pek çok kişi "yapay zekânın kışı" olarak adlandırılan bu dönemde teknolojiye olan ilginin azaldığını gözlemledi. Bununla birlikte akademik çevrelerde istatistiksel yaklaşımlar üzerinde çalışılmaya devam edildi.

### 1980’ler: Yeni Yaklaşımlar

1980'lerde geri yayılım (backpropagation) algoritması tanıtıldı. Geri yayılım (backpropagation) bir yapay sinir ağının öğrendiği hataları düzeltmek için kullandığı bir yöntemdir. Ağın tahmin ettiği sonuç ile gerçek sonuç arasındaki farkı bulur ve bu farkı kullanarak ağın içindeki bağlantıların (ağırlıkların) daha doğru olmasını sağlar. Yani hatayı geri gönderip ağırlıkları güncelleyerek ağın daha doğru sonuçlar üretmesini sağlar. Bu yöntem, sinir ağlarının eğitimi için kritik bir adımdı ve daha karmaşık problemlerin çözülmesini sağladı. IBM, bu dönemde yapay zekâ projelerine önemli yatırımlar yaptı ve bu alandaki araştırmaları destekledi​.

### 1990’lar: Veri Madenciliği ve İstatistiksel Öğrenme

1990’lar veri madenciliği ve istatistiksel öğrenme teorilerinin ön planda olduğu bir dönemdir. Destek Vektör Makineleri (SVM) gibi algoritmalar büyük veri kümelerini analiz etmede etkili bir araç olarak popülerleşti. Ayrıca bilgisayarların işlem gücü önemli ölçüde arttı ve makine öğrenmesi uygulamaları için daha geniş bir alan sağlandı​.

### 2000’ler: Derin Öğrenme ve Büyük Veri

Bu dönemde büyük veri ve GPU teknolojilerindeki gelişmeler makine öğrenmesine büyük bir ivme kazandırdı. 2012’de sinir ağlarını kullanarak görsel tanıma yarışmasını kazanan AlexNet , derin öğrenmenin gücünü gösterdi ve bu alanın hızla büyümesine yol açtı. IBM, bu dönemde Watson projesiyle dikkat çekti. Watson, Jeopardy yarışmasını kazanarak makine öğrenmesinin gerçek dünyadaki potansiyelini gözler önüne serdi.

IBM Watson bir yapay zekâ sistemidir. Watson , IBM tarafından tasarlanıp geliştirilen bir [süper bilgisayardı. Bu gelişmiş bilgisayar yapay zekâyı (](https://www.techtarget.com/whatis/definition/supercomputer)[AI](https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence) ), otomasyonu ve karmaşık analitik yeteneklerini bir araya getirerek "soru cevaplama" makinesi olarak en iyi performansı sunuyordu. Watson özellikle doğal dil işleme ve makine öğrenmesi kullanarak insan dilini anlamaya ve doğru cevaplar vermeye çalışır. Doğru cevapları metin olarak verir. En ünlü başarısını 2011’deki “Jeopardy!” adlı bilgi yarışmasında gösterdi. Watson insan rakiplerini yenerek soruları anlamada ve doğru yanıtlar vermede ne kadar güçlü olduğunu kanıtladı. Bu, Watson’ın sadece bilgi depolamakla kalmayıp insan gibi düşünebildiğini ve karmaşık soruları çözebildiğini gösterdi.

AlexNet, 2012 yılında görsel tanıma (image recognition) alanında büyük bir yenilik getiren bir yapay zekâ modelidir. Bu model sinir ağları kullanarak resimleri tanımada çok yüksek doğruluk oranlarına ulaşmıştır. Özellikle derin öğrenme tekniklerini kullanarak bilgisayarlara resimlerdeki nesneleri ve özellikleri nasıl tanıyacaklarını öğretmiştir. Örneğin AlexNet bir fotoğrafın içinde kedi, köpek, araba gibi nesneleri tanımlayabilir. Bu bilgisayarların görselleri insanlar gibi anlamaya başlaması anlamına gelir. Kısacası AlexNet , görüntüleri anlamak ve sınıflandırmak için geliştirilen derin öğrenme tabanlı bir modeldir ve modern yapay zekâ uygulamalarında önemli bir dönüm noktasıdır.

### Günümüz

Makine öğrenmesi günümüzde sağlık teşhisi, özerk araçlar, sesli asistanlar ve öneri sistemleri gibi birçok alanda kullanılmaktadır. Teknolojik gelişmelerle birlikte bu alanda daha yenilikçi ve etkili yöntemler geliştirilmeye devam edilmektedir.

## MAKİNE ÖĞRENMESİ HANGİ ALANLARDA KULLANILIR?

Makine öğrenmesi, birçok sektörde devrim niteliğinde yeniliklere yol açan bir teknoloji olarak geniş bir kullanım alanına sahiptir. İşte bu alanlardan bazıları ve örnekler:

### Sağlık

Makine öğrenmesi hastalıkların daha hızlı teşhis edilmesi, tedavi planlarının kişiselleştirilmesi ve ilaç keşfi gibi sağlık sektöründe önemli yenilikler sağlamaktadır. Örneğin:

* ***Kanser Tespiti:*** Google Health , makine öğrenmesi kullanarak akciğer kanseri gibi hastalıkları tespit etmek için görüntüleme tekniklerini analiz eden sistemler geliştirdi. Bu sistemler, röntgen ve BT taramalarını kullanarak kanserli hücreleri daha hızlı ve doğru bir şekilde tespit ediyor.
* ***Kişisel Tedavi Planları:*** IBM'in Watson for Health , genetik veriler ve hasta geçmişine dayalı tedavi önerileri geliştirebiliyor, böylece hastaların daha hedeflenmiş tedavi almalarını sağlıyor.

### Finans

Makine öğrenmesi finans sektöründe risk analizi, dolandırıcılık tespiti ve portföy yönetimi gibi alanlarda kullanılır. Örneğin:

* ***Dolandırıcılık Tespiti****:* PayPal ve diğer finansal kuruluşlar, ödeme işlemleri sırasında şüpheli faaliyetleri tespit etmek için makine öğrenmesi algoritmalarını kullanır.
* ***Portföy Yönetimi****:* Makine öğrenmesi, algoritmaların ticaret stratejileri oluşturmasını ve yatırımcıların kararları hızla almasını sağlar. BlackRock gibi şirketler yatırım portföylerini optimize etmek için bu tür teknolojileri kullanıyor.

### Otomotiv

Makine öğrenmesi otonom araçlar ve sürüş asistanları gibi teknolojilerle otomotiv sektöründe devrim yaratıyor. Örneğin:

* ***Otonom Araçlar****:* Tesla, araçlarının kendi kendine sürüş yapabilmesi için derin öğrenme algoritmalarını kullanıyor. Bu araçlar çevrelerini analiz ederek güvenli bir şekilde yol alabiliyor. Google’ın otonom araçları geliştiren alt şirketi olan Waymo , yapay zekâ ve makine öğrenmesi kullanarak araçları otonom hale getiriyor.
* ***Sürüş Asistanları****:* Audi ve BMW gibi markalar, sürücülerin dikkatini çeken uyarılar ve güvenlik sistemleri için makine öğrenmesi kullanmaktadır.

### Perakende ve E-Ticaret

Makine öğrenmesi müşteri deneyimini kişiselleştirmek ve satışları artırmak için yaygın olarak kullanılmaktadır. Örneğin:

* ***Kişiselleştirilmiş Öneriler****:* Amazon, kullanıcılarının önceki alışveriş verilerine dayalı olarak ürün önerileri sunar. Bu öneriler kullanıcıların alışveriş alışkanlıklarına göre optimize edilmiştir.
* ***Stok Yönetimi***: Walmart ve diğer büyük perakendeciler, talep tahminlerini yapmak ve envanter yönetimini iyileştirmek için makine öğrenmesi algoritmalarını kullanıyor.
* ***Pazarlama Çözümü****:* Shopify , e-ticaret siteleri için yapay zekâ ve makine öğrenmesiyle kişiselleştirilmiş pazarlama çözümleri sunar.

### Lojistik ve Ulaşım

Makine öğrenmesi güzergâh optimizasyonu, trafik analizi ve araç takip sistemlerinde de kullanılıyor. Örneğin:

* ***Trafik Tahminleri****:* Google Maps , trafik sıkışıklıklarını tahmin etmek ve alternatif rotalar önermek için makine öğrenmesi ve gerçek zamanlı veri analizi yapar.
* ***Güzergâh Optimizasyonu****:* UPS, teslimat rotalarını optimize etmek ve yakıt tüketimini azaltmak için makine öğrenmesi kullanıyor. DHL, makine öğrenmesi ve yapay zekâ teknolojileri kullanarak, lojistik süreçlerde verimliliği artırmaya yönelik projeler geliştirmiştir. Özellikle paket takibi, rota planlaması ve depolama yönetimi gibi alanlarda makine öğrenmesi algoritmaları sayesinde operasyonel maliyetleri düşürmekte ve teslimat sürelerini iyileştirmektedir.

### Eğitim

Makine öğrenmesi eğitim materyallerini kişiselleştirme, öğrenci performansını izleme ve öğretim araçlarını geliştirme gibi alanlarda kullanılıyor. Örneğin:

* ***Kişiselleştirilmiş Öğrenme****:* Duolingo, öğrencilerin dil öğrenme süreçlerini takip ederek, kişiselleştirilmiş dersler ve geri bildirimler sunar. Khan Academy, öğrenci performansını izleyip, kişiselleştirilmiş eğitim materyalleri sunmak için yapay zekâ tabanlı araçlar kullanır.
* ***Öğrenci Performansını İzleme****:* Öğretim yönetim sistemleri öğrencilerin başarı düzeylerini analiz eder ve gerektiğinde müfredatta değişiklikler önerir.
* ***Çocuk Eğitimi****:* Makine öğrenmesi oyuncaklarda da kullanılıyor. Bazı oyuncaklar çocukların verdiği yanıtlara göre tepki vererek bir diyalog kurabiliyor. Örneğin akıllı oyuncaklar, sesli komutları anlayıp tepki vererek çocukların dil öğrenme süreçlerine yardımcı olabilir. Fisher-Price gibi markalar bu teknolojiyi sesli yanıtlar ve etkileşimli öğrenme için kullanıyor.

### Hukuk

Makine öğrenmesi hukuk sektöründe belge incelemesi, sözleşme analizi ve davaların tahmin edilmesinde kullanılır. Örneğin:

* ***Belgelerin Otomatik İncelenmesi****:* ROSS Intelligence, avukatların davaları inceleme sürecini hızlandırmak için makine öğrenmesi kullanır. Algoritmalar hukuk metinlerini analiz ederek en uygun hukuk tavsiyelerini sunar.
* ***Dava Sonucu Tahmini****:* Lex Machina, davaların geçmiş verilerini analiz ederek gelecekteki davaların sonucunu tahmin etmeye yönelik araçlar geliştirir.

Bu örnekler makine öğrenmesinin hayatımızdaki birçok sektörü dönüştürmeye nasıl yardımcı olduğunu göstermektedir. Teknoloji ilerledikçe bu uygulamaların daha fazla alanda yaygınlaşması ve etkilerini daha da artırması beklenmektedir.

## MAKİNE ÖĞRENMESİNDE TEMEL KAVRAMLAR

### Veri (Data) nedir?

İşlenmemiş gerçek enformasyon parçacığına verilen isimdir. Bilgisayar ortamında bulunan bilgilerin, programlar tarafından işlenebilmesini sağlamak amacı ile derlenmiş ve formüle edilmiş şekline veri denir.Bu kavram genellikle enformasyon alanında kullanılan bir terimi ifade etmek için kullanılır.

**Şekil 2**



Tek başına anlam ifade etmez ve kullanılmaz. Veri; ölçüm, deney, gözlem, sayım ya da araştırma yolu ile elde edilir. Bunun yanında enformasyona ve bilgiye temel oluşturan gruplandırmaya ve analiz edilmeye gereksinim duyulan ham bilgi olarak tanımlanır.

**Veri Madenciliği:** Büyük veri kümelerini işlemek ve keşfetmek üzere analiz alanında kullanılan bilgisayar destekli bir tekniktir. Kuruluşlar, veri madenciliği araçları ve yöntemleriyle, verilerindeki gizli düzenleri ve ilişkileri keşfedebilir. Veri madenciliği, ham verileri pratik bilgilere dönüştürür.

### Veri Ön İşleme (Data Preprocessing) nedir?

Veri ön işleme; veri madenciliği modelleri kurulmadan önce veri seti üzerinde yapılan bir takım düzeltme, eksik veriyi tamamlama, tekrarlanan verileri kaldırma, dönüştürme, bütünleştirme, temizleme, normalleştirme, boyut indirgeme vb. işlemlerdir.

**Veri Ön İşleme İşlemleri**

1. ***Veri Temizleme:*** Aykırı verilerin (outlier) temizlenmesi, gürültü verilerinin düzeltilmesi, tutarsızlıkların giderilmesi, eksik değerlerin doldurulması
2. ***Veri Entegrasyonu:*** Veri tabanı, veri küpü veya dosya entegrasyonu.
3. ***Veri Dönüşümü (Data Transformation):*** Normalizasyon (Input değeri indirgemek anlamına gelir. Veriler arasında farklılığın çok fazla olduğu durumlarda verileri tek bir düzen içerisinde ele almaktır.)

* ***Veri Azaltma (Data Reduction):*** Hacmi küçültme, veri sıkıştırma. Önemsiz niteliklerin kaldırılması

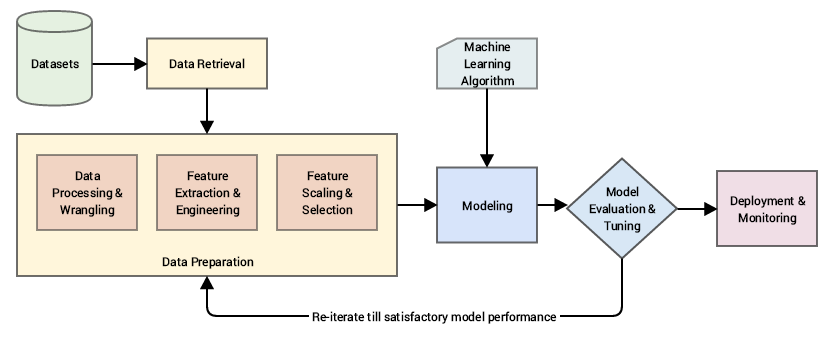
1. ***Veri Ayrıklaştırma:*** Verinin azaltılması.

### Özellik Mühendisliği (Feature Engineering) nedir?

Özellik mühendisliği, mevcut veri setindeki bilgileri kullanarak yeni ve daha bilgilendirici özellikler oluşturma sürecidir. Bu yeni özellikler, model performansını artırmak ve daha iyi sonuçlar elde etmek amacıyla kullanılır. Özellik mühendisliği, veri bilimi ve makine öğrenmesi projelerinde kritik bir aşamayı temsil eder.

Özellik mühendisliğinin önemi de şu şekildedir: Veri setindeki özellikler kullanılan tahmine dayalı modelleri ve elde edilen sonuçları doğrudan etkiler. Verilerinizin doğasında bulunan yapıları tanımlayan özelliklere ihtiyaç vardır. Çünkü daha iyi özellikler esneklik demektir. Daha iyi özellikler, daha basit modeller ve daha iyi sonuçlar anlamına gelir.

**Şekil 3**



### Model ve algoritma nedir?

|  |  |
| --- | --- |
| **ML Model Türü** | **Kullanım Alanları** |
| Doğrusal regresyon/sınıflandırma | Sayısal verilerdeki örüntüler, finansal tablolar gibi |
| Grafik modeller | Fraud tespiti veya duygusal farkındalık |
| Karar ağacı | Sonuçları tahmin etmek |
| Derin öğrenme sinir ağları | Computer Vision, NLP ve dahası. |

**Tablo 1: Model türleri ve kullanım alanları**

Bir makine öğrenimi modeli, örüntüleri bulmak veya tahminler yapmak için büyük bir veri grubunu tarayan bir algoritmanın ifadesidir. Verilerle desteklenen makine öğrenimi modelleri, yapay zekânın matematiksel motorlarıdır.

Makine öğrenmesi algoritmaları, kişilerin, karmaşık veri kümelerini keşfetmesi, analiz etmesi ve bunlarda anlam bulmasına yardımcı olan kod parçacıklardır. Her algoritma, bir makinenin belirli bir hedefi gerçekleştirmek için izleyebileceği sınırlı ve belirli, adım adım ilerleyen yönerge kümesidir. Makine öğrenmesi modelinin hedefi, kişilerin tahmin yapmak veya bilgileri kategorilere ayırmak için kullanabileceği desenler oluşturmak veya keşfetmektir.

Algoritmaların 3 türü vardır: Denetimli, Denetimsiz ve Pekiştirmeli

### Eğitim, doğrulama ve test nedir?

1. ***Eğitim***

Verilere dayalı tahminler veya kararlar almak için bir makine öğrenimi algoritmasına öğretim sürecidir. Bir makine öğrenimi modelinin eğitim süreci, akıllı sistemlerin üzerine inşa edildiği temeldir. Kullanılacak makine öğrenimi algoritması, eğitim kümesini kullanarak tahminlerde bulunmayı veya verileri kategorilere ayırmayı öğrenir. Bu aşamada model, eğitim veri kümesiyle en iyi şekilde eşleşmesi için iç ayarlarını iyileştirir.

1. ***Doğrulama***

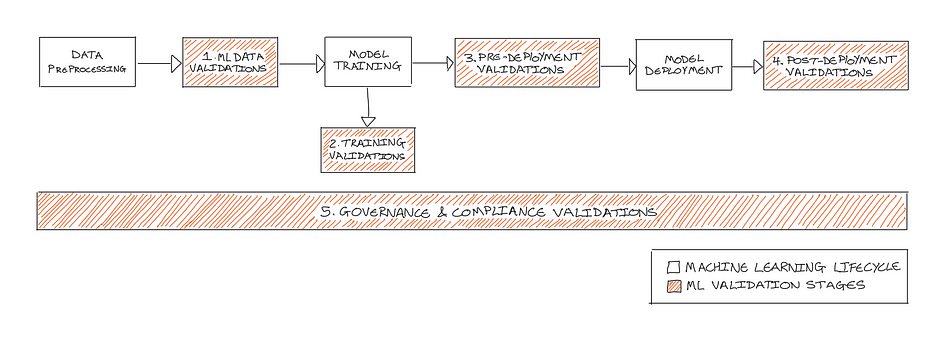
Eğitilmiş bir modelin performansını değerlendirmemize yardımcı olan sürece Model Doğrulama denir. Bu, yeni veya görülmemiş veriler üzerinde makine öğrenimi modelinin performansını doğrulamamıza yardımcı olur. Ayrıca, modelin amaçlanan amacına ulaştığını doğrulamamıza yardımcı olur.

Makine öğreniminin olasılıksal doğası nedeniyle, makine öğrenimi sistemlerini geleneksel yazılımlarla aynı şekilde (yani birim testleri, entegrasyon testi vb.) test etmek zordur. Ek olarak, modelde veya verilerde bir hata tespit edilirse, çözüm yalnızca bir düzeltmeyi uygulamak olamaz. Yine, bunun nedeni bir makine öğrenmesi modeli etrafındaki değişen ortamlar ve yeniden eğitme ihtiyacıdır. Çözüm yalnızca bir model düzeltmesiyse, model bir sonraki yeniden eğitildiğinde veya veriler güncelleştirildiğinde, düzeltme kaybolur ve artık hesaba katılmaz. Bu nedenle, belirli model davranışlarını ve veri kalitesini kontrol etmek için model doğrulamaları uygulanmalıdır. Üretime alınmadan önce hem makine öğrenimi sistem kalitesini doğrulamak için makine öğrenimi yaşam döngüsünün tüm aşamalarında testler yapılmalıdır.

Aşağıda **Şekil 3’te** gösterildiği gibi, makine öğrenimi doğrulamasının 5 temel aşaması tanımlanmıştır:

1. Makine öğrenimi veri doğrulamaları
2. Eğitim doğrulamaları
3. Dağıtım öncesi doğrulamalar
4. Dağıtım sonrası doğrulamalar
5. Yönetişim ve uyumluluk doğrulamaları

**Şekil 4**



* ***Test***

[Makine öğrenimini test etmek zordur](https://medium.com/@Eyaltra/d53623d32797#78cd) çünkü doğası gereği olasılıksaldır ve çeşitli verileri ve dinamik gerçek dünya koşullarını hesaba katması gerekir. Makine öğrenimi modellerinde test yaparken güvenilirlik, sağlamlık ve adalet kavramları ile ilgilenmelidir.

Test, makine öğrenmesi modellerinin doğruluğunu veya başarısını ölçmek için kullanılan bir veri setiyle gerçekleştirilir. Test aşaması, aşağıdaki sorulara yanıt arar:

* Model, eğitim sırasında öğrendiği bilgileri yeni verilere uygulayabiliyor mu?
* Model aşırı öğrenme (overfitting) veya yetersiz öğrenme (underfitting) sorunları yaşıyor mu?
* Model, pratik uygulamalarda beklendiği gibi performans gösterebilir mi?

Test sürecini önemli hale getiren unsurlarda overfitting’in tespiti, hiper parametrelerin değerlendirilmesi ve gerçek dünya performansı vardır.

\* Değerlendirme ile test etme aynı şey değildir. Farkı bilmek önemlidir.

|  |  |  |
| --- | --- | --- |
| **Yüz**  **(Aspect)** | **Model Değerlendirmesi**  **(Model Evaluation)** | **Model Testi**  **(Model Testing)** |
| Odak | Genel performans | Detaylı bileşen analizi |
| Ölçüm | Doğruluk, Kesinlik, Kök Ortalama Kare, Geri Çağırma vb. | Kod, Veri ve Model davranışı |
| Nesnel | Performansı izleme | Kusurları/güvenlik açıklarını belirleyin ve düzeltin |
| Sınırlama | Başarısızlığın nedenlerini açıklamıyor | Daha fazla kaynak ve zaman gerektirir |

**Tablo 2: Model değerlendirmesi ve testi**

## NASIL ÇALIŞIR?

Makine Öğrenmesi; makinelerin çeşitli veri ve algoritmalar kullanarak, problem karşısında bir insan gibi çözüm üretmesi ya da kademeli bir şekilde çözüme ulaşmasını sağlayan bir bilgisayar bilim dalıdır.

Makine öğrenmesinin temelinde “yapay zekâ” vardır. Yapay zekânın çalışma şekline göre makine öğrenmesi farklı şekillerde gerçekleşebiliyor. Ancak makine öğrenmesi temel olarak şu 3 adımda gerçekleşmektedir;

* Karar Süreci: Makine öğrenimi algoritmaları, genellikle bir tahmin veya sınıflandırma yapmak için kullanılır. Makine öğrenimi modelleri, etiketlenebilen ya da etiketlenemeyen girdi verileri ile çalışır. Algoritma, bu veriler üzerinde bir desen bularak tahmin üretir. Örneğin, bir makine öğrenimi modeli, bir müşterinin gelecekte bir ürünü satın alıp almayacağını tahmin edebilir. Burada algoritmanın karar süreci, girdi verileri ile öğrendiği desenler doğrultusunda gerçekleşir. Makine öğrenimi sayesinde algoritmalar sürekli olarak yeni verilerden öğrenir ve performanslarını geliştirir.
* Hata İşlevi: Makine öğrenimi algoritmalarının tahminlerinin doğruluğunu ölçen bir başka önemli bileşen hata işlevidir. Hata işlevi, modelin verdiği tahminin ne kadar doğru olduğunu değerlendirir. Bilinen örnekler varsa, hata işlevi modelin performansını değerlendirir. Örneğin, makine öğrenimi algoritması bir veri kümesi üzerinde çalışıyorsa ve tahmin edilen sonuçlar bilinen sonuçlarla karşılaştırılabiliyorsa, hata işlevi bu farkı değerlendirir. Hata işlevi, makine öğrenmesi sürecinde algoritmanın ne kadar doğru çalıştığını anlamamıza yardımcı olur.
* Model Optimizasyon Süreci***:***[**Makine öğrenimi**](https://www.datacamp.com/blog/what-is-machine-learning)algoritmalarının bir diğer önemli bileşeni de model optimizasyon sürecidir. Bu süreç, modelin eğitim verilerindeki hata oranını en aza indirmek için çalışır. Makine öğrenimi algoritması, modelin ağırlıklarını sürekli olarak güncelleyerek doğruluğu artırır. Bu işlemi tekrar eden algoritma, her seferinde daha doğru tahminler yapmaya başlar. Makine öğrenimi modelleri, bir doğruluk eşiğine ulaşana kadar bu optimize etme sürecini devam ettirir. Bu sayede, algoritma verilerle daha uyumlu hale gelir ve daha doğru tahminler üretir.

## MAKİNE ÖĞRENMESİ TÜRLERİ NELERDİR?



**Şekil 5**

### Denetimli Öğrenme (Supervised Learning)***Nedir?***

Denetimli öğrenme, makine öğrenmesinin en yaygın kullanılan alt dallarından biridir ve etiketlenmiş veri setleri kullanarak öğrenme süreçlerini gerçekleştirir. Bu yöntem, eğitim verileri üzerinden modelin girdiler ve çıktılar arasında bir ilişki kurmasını sağlar. Böylece model, yeni ve daha önce karşılaşmadığı veri noktalarını doğru bir şekilde tahmin edebilir veya sınıflandırabilir.

**Çalışma Prensibi**

Denetimli öğrenmede, her veri noktası bir giriş (X) ve bu girişle ilişkili doğru bir çıktı (Y) değeri içerir. Algoritma, bu ilişkiden bir fonksiyon öğrenerek girdilerle çıktılar arasında bir haritalama oluşturur. Bu süreç, iki temel alt kategoriye ayrılır:

* *Sınıflandırma:* Veri noktalarını belirli kategorilere ayırır (örneğin, bir e-postanın spam olup olmadığını tahmin etmek).
* *Regresyon:* Sürekli değişkenlerin tahmin edilmesini sağlar (örneğin, bir evin piyasa değerini tahmin etmek).

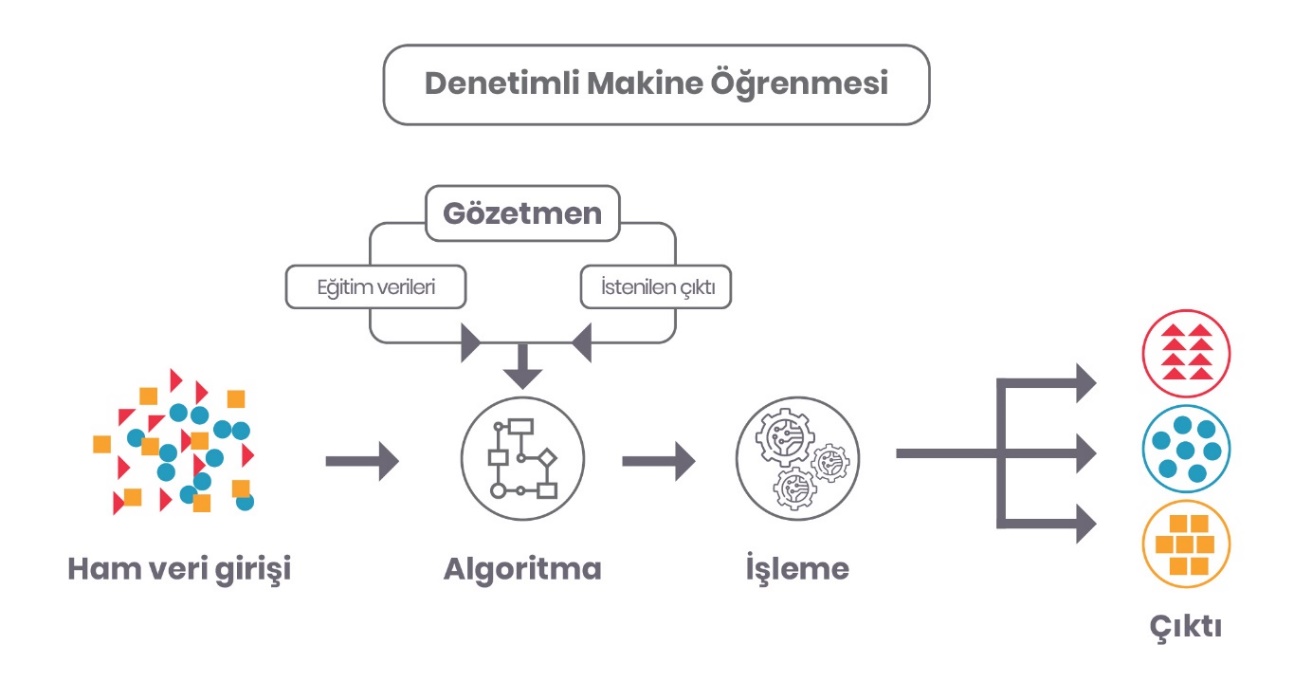
**Algoritmalar ve Kullanım Alanları**

1. *Karar Ağaçları ve Rastgele Ormanlar:* Karar yapıları üzerinden sınıflandırma veya regresyon işlemleri yapar.
2. *Destek Vektör Makineleri (SVM):* Veri noktalarını düzlem üzerinde ayıran bir hiper düzlem oluşturur.
3. *Sinir Ağları:* Daha karmaşık, çok katmanlı ilişkileri modellemek için kullanılır.
4. *Doğrusal ve Lojistik Regresyon*: Doğrudan ilişkiler kurarak tahmin veya sınıflandırma yapar.

Denetimli öğrenme modelleri, tıp (hastalık teşhisi), finans (kredi risk analizi) ve pazarlama (müşteri segmentasyonu) gibi pek çok alanda geniş bir uygulama alanına sahiptir.

**Önemli Hususlar**

1. *Veri Kalitesi:* Etiketlenmiş veriler doğru ve kaliteli olmalıdır, çünkü modelin başarısı büyük ölçüde veri setine bağlıdır.
2. *Aşırı Uyum (Overfitting):* Model, eğitim verisine aşırı uyum sağlayarak yeni veri üzerindeki performansını düşürebilir. Bu durum, daha iyi genelleştirme teknikleri ile önlenebilir.
3. *Hesaplama Gücü:* Bazı denetimli öğrenme algoritmaları yoğun hesaplama kaynaklarına ihtiyaç duyar.



**Şekil 6: Denetimli Makine Öğrenmesi**

Denetimsiz Öğrenme (Unsupervised Learning) Nedir?  
Denetimsiz öğrenme, etiketsiz veriler arasındaki gizli yapı veya örüntüyü bulmayı amaçlayan bir makine öğrenmesi yöntemidir. Bu yöntemde, elimizdeki veriler etiketlenmemiş olduğundan, herhangi bir sınıflandırma bilgisi olmadan verilerin arasındaki ilişkileri analiz etmeye çalışırız. Gözetimsiz öğrenme, bağımlı değişkenin olmadığı ve yalnızca girdi değişkenlerinin yer aldığı problemlerde kullanılır. Veriler üzerinde istatistiksel ve matematiksel teknikler uygulanarak gizli desenler keşfedilir, veriler gruplandırılır ve benzerlikler veya farklılıklar belirlenir.

Gözetimsiz öğrenme, özellikle veri analizi, keşfi ve ön işleme süreçlerinde oldukça sık kullanılan bir yöntemdir. Bu yöntem, verilerin daha iyi anlaşılmasını ve görselleştirilmesini sağlarken, diğer makine öğrenmesi tekniklerine de temel oluşturur. Örneğin, elimizde çeşitli renklerde lego parçalarından oluşan bir yığın olduğunu düşünelim. Bu lego parçalarını renklerine göre ayırarak grupladığımızda, her bir grup "küme" olarak adlandırılır. Gözetimsiz öğrenme, bu tür gruplamaları yapmak için kullanılır ve yeni bir veri geldiğinde hangi gruba ait olduğunu belirlemeye yardımcı olur.

**Gözetimsiz Öğrenme Yöntemleri**

* ***K-Ortalamalar Yöntemi (K-Means)***  
  K-Ortalamalar, etiketsiz veriler üzerinde kümeleme yapmak için kullanılan basit ve popüler bir yöntemdir. Veriler, benzer özelliklerine göre gruplandırılır ve her bir grup bir küme olarak adlandırılır. Algoritma, veriler arasında doğal olarak var olan grupları keşfetmeyi ve bu gruplar arasındaki örüntüleri belirlemeyi hedefler.

#### K-Means Algoritmasının Çalışma Adımları:

1. İlk olarak, K adet küme sayısı belirlenir.
2. Kadar rastgele merkez noktası seçilir.
3. Her bir veri noktası, kendisine en yakın olan küme merkezine atanır.
4. Atama işlemi yapıldıktan sonra kümelerin yeni merkez noktaları hesaplanır.
5. Bu işlem, kümelerin konumları sabitlenene kadar tekrarlanır.

K-Means, pazarlama ve müşteri segmentasyonu, görüntü işleme, biyoinformatik ve finansal analiz gibi çeşitli alanlarda yaygın olarak kullanılır.

* *Hiyerarşik Kümeleme Analizi (Hierarchical Cluster Analysis)*  
  Hiyerarşik kümeleme, gözlemler arasındaki benzerliklere dayanarak verileri alt kümelere ayıran bir yöntemdir. Sonuçlar, dendrogram adı verilen ağaç yapısında görselleştirilir. İki temel yaklaşıma sahiptir:

*Birleştirici (Agglomerative) Yaklaşım:* Aşağıdan yukarıya bir yaklaşımdır; her bir gözlem başlangıçta ayrı bir küme olarak kabul edilir ve kümeler birbirine benzerliklerine göre birleştirilir.

*Bölücü (Divisive) Yaklaşım:* Yukarıdan aşağıya bir yaklaşımdır; başlangıçta tüm gözlemler tek bir küme olarak kabul edilir ve her adımda kümeler bölünür.

Hiyerarşik kümeleme; biyoinformatik, pazarlama, coğrafi bilgi sistemleri (CBS), veri görselleştirme ve finansal analiz gibi birçok alanda kullanılır.

* Temel Bileşen Analizi (Principal Component Analysis, PCA)  
  PCA, çok boyutlu bir veri setini daha az boyutlu bir uzaya indirgemeyi amaçlar. Bu yöntem, korelasyonlu değişkenleri temel bileşenler adı verilen yeni değişkenlere dönüştürür. İlk temel bileşen, verideki en büyük varyansı temsil eder; diğer bileşenler ise azalan varyans değerine göre sıralanır. PCA, özellikle yüksek boyutlu veri setlerinde gürültüyü azaltma, veri sıkıştırma ve veriyi görselleştirme için kullanılır.

#### Kullanım Alanları:

* Veri ön işleme ve boyut azaltma
* Görüntü işleme
* Finansal veri analizi
* Genetik veri analizi

### Pekiştirmeli Öğrenme (Reinforcement Learning — RL) Nedir?

Pekiştirmeli öğrenme (reinforcement learning — RL), makine öğrenimi alanının önemli bir dalıdır. Bu yaklaşım, insanların ve hayvanların deneyimlerinden öğrenme süreçlerine benzer şekilde, algoritmaların ve yapay zekâ sistemlerinin ödül ve ceza mekanizmalarını kullanarak öğrenmelerini sağlar. Pekiştirmeli öğrenme algoritmaları, önceden etiketlenmiş verilere dayalı değil, ancak doğrudan etkileşimlerle ve denemelerle öğrenirler.

***Pekiştirmeli Öğrenme Unsurları***

*Ajan (Agent):* Çevreden bilgi alıp aksiyonlar gerçekleştiren yapıdır. Bir robot, algoritma ya da yazılım sistemi ajanı temsil edebilir.

*Çevre (Environment):* Ajanın eylemlerine geri bildirim sağlayan ortamı ifade eder.

*Durum (State):* Ajanın, çevresindeki anılık konumunu ya da durumunu tanımlar.

*Aksiyon (Action):* Ajanın belirli bir durumda aldığı karar ya da eylemdir.

*Ödül (Reward):* Ajanın bir eylem sonrasında elde ettiği başarı (pozitif) ya da başarısızlık (negatif) sinyalleridir.

***Pekiştirmeli Öğrenmenin Çalışma Mekanizması***

Pekiştirmeli öğrenme, ajan-çevre etkileşiminin zaman adımları boyunca tekrarlanmasını içerir. Süreç şu şekilde işler:

* Ajan, çevredeki mevcut durumu (state) algılar.
* Bir politika doğrultusunda bir eylem (action) seçer ve uygular.
* Çevre, bu eyleme göre bir sonraki durumu ve bir ödül (reward) döndürür.
* Ajan, ödül bilgisini kullanarak politikasını iyileştirir.

***Değer Tabanlı Yöntemler (Value-Based Methods):***

* Amaç, durumların ya da eylemlerin değerlerini tahmin ederek bir politika geliştirmektir.
* Örneğin, Q-Learning ve Deep Q-Network (DQN) algoritmaları bu gruba girer.

***Politika Tabanlı Yöntemler (Policy-Based Methods):***

* Doğrudan bir politika öğrenmeyi hedefler. Politika genellikle bir olasılık dağılımı olarak modellenir.
* Örnek: REINFORCE, Proximal Policy Optimization (PPO).

***Değer-Politika Hibrit Yöntemler (Actor-Critic Methods):***

* Hem bir değer fonksiyonu hem de bir politika öğrenir.
* Örnek: Advantage Actor-Critic (A2C), Deep Deterministic Policy Gradient (DDPG).

**Pekiştirmeli Öğrenmenin Kullanım Alanları Nedir?**

### 1*. Robotik*

* Manipülasyon: Robotların nesneleri toplaması, taşıması ve yerleştirmesi gibi görevlerde optimize edilmiş eylemleri öğrenmesi.
* Navigasyon: Robotların engellerden kaçınarak hedefe ulaşmasını sağlayan öğrenme sistemleri.
* Endüstriyel Otomasyon: Fabrikalarda robotların verimli üretim süreçleri geliştirmesi.

### 2*. Oyunlar*

* Satranç, Go ve Poker: DeepMind’ın AlphaGo'su gibi sistemler bu tür oyunlarda insanüstü başarıya ulaşmıştır.
* Video Oyunları: Oyun içi strateji geliştirme ve oynama becerilerini optimize etmek için kullanılır.

### 3*. Otonom Araçlar*

1. Otonom sürüş sistemlerinde pekiştirmeli öğrenme, çevre algısı ve güvenli karar alma mekanizmalarını geliştirmek için kullanılır.
2. Araç Kontrolü: Trafik kurallarına uyarak, yol koşullarına adapte olma.

### 4*. Finans*

* Portföy Optimizasyonu: Yatırım araçlarını en iyi şekilde dağıtmak için RL kullanılır.
* Ticaret Stratejileri: Karmaşık piyasa dinamiklerine uyum sağlayarak ticaret kararları alınması.

### 5*.Sağlık*

* Tedavi Planlaması: Hastalar için kişiselleştirilmiş tedavi süreçleri geliştirmek.
* İlaç Keşfi: İlaç moleküllerinin davranışlarını modellemek ve optimize etmek.
* Robotik Cerrahi: Pekiştirmeli öğrenme, cerrahi robotların karmaşık prosedürleri gerçekleştirmesine yardımcı olur.

### 6*.Doğal Dil İşleme (NLP)*

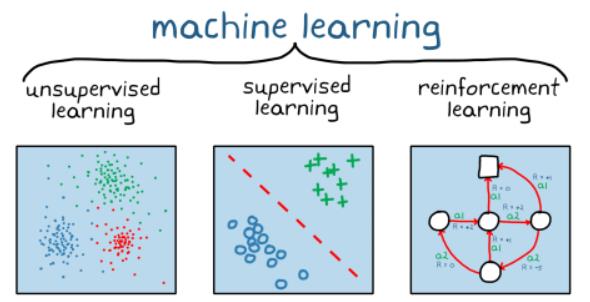
1. Konuşma Tanıma ve Çeviri: Daha etkili dil modelleri geliştirmek.
2. Sohbet Robotları: Kullanıcılarla etkileşimde daha etkili yanıtlar sunmak.

### 7*. Enerji Yönetimi*

1. Akıllı Şebekeler: Enerji talebine göre üretim ve tüketim dengesini optimize etme.
2. Bina Enerji Yönetimi: Isıtma, soğutma ve aydınlatma sistemlerini optimize etme.

### 8*. Reklamcılık ve Öneri Sistemleri*

1. Dinamik Reklamcılık: Kullanıcı davranışlarını öğrenerek en uygun reklamları sunmak.
2. Öneri Sistemleri: Kullanıcının geçmiş tercihlerine dayanarak içerik önerileri yapmak.



**Şekil 7: Makine öğrenmesi türleri**

## MAKİNE ÖĞRENMESİNDE ALGORİTMALAR VE MODELLER

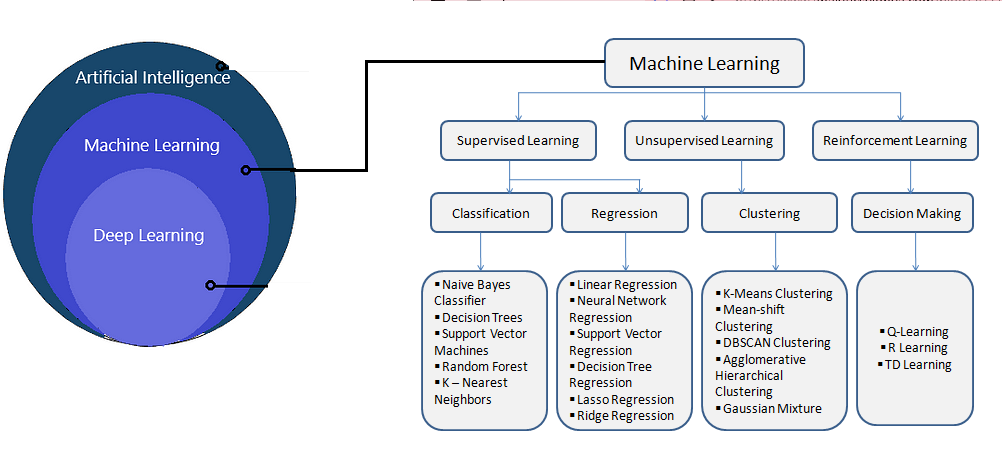
### GİRİŞ

Makine öğrenmesi, verilerden modeller oluşturarak, bilgisayarların kalıpları anlamasını ve tahminlerde bulunmasını sağlayan bir yöntemler bütünüdür. Makine öğrenimi algoritmaları, bu süreci yöneten motorlardır ve bir veri setini model haline dönüştürmek için kullanılan temel araçlardır. Hangi algoritmanın en iyi şekilde çalıştığı, analiz edilen verinin türüne, mevcut kaynaklara, verilerin niteliğine ve hedeflenen sonuca bağlı olarak değişir.

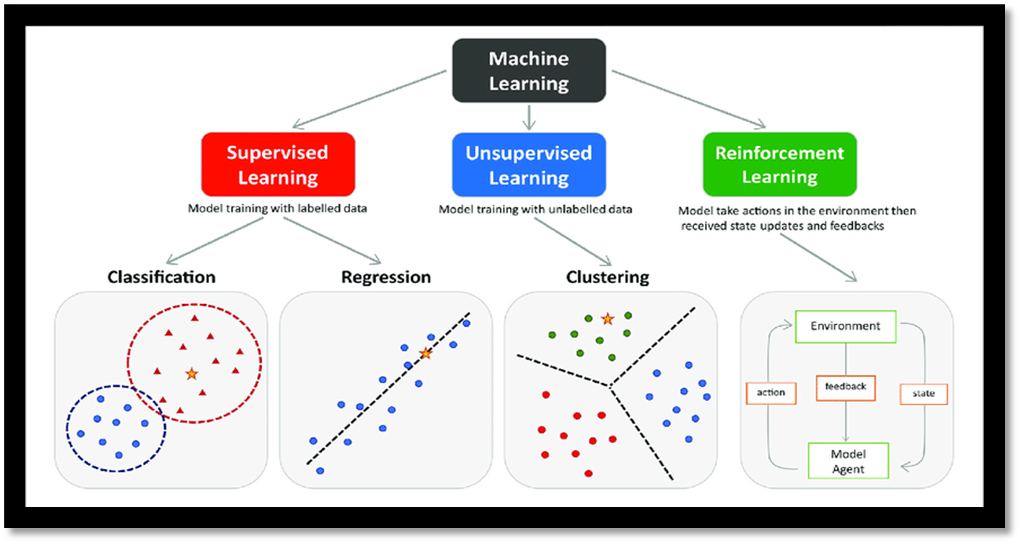
Makine öğrenimi, bilgisayarların açık programlamaya gerek duymadan veriler üzerinden öğrenmesini, kalıplar keşfetmesini ve tahminlerde bulunmasını mümkün kılar. Bu algoritmalar, yapay zekânın temelini oluşturarak, görüntü tanıma, konuşma tanıma, doğal dil işleme, öneri sistemleri, dolandırıcılık tespiti, otonom araçlar gibi çeşitli alanlarda uygulanmaktadır. Ayrıca, daha doğru tahminler yapmak için algoritmalar büyük veri kümeleri üzerinde eğitildiğinde, karar verme süreçlerini iyileştirme ve daha verimli çözümler üretme imkânı sağlar.

Makine öğrenimi algoritmaları, genel olarak üç ana kategoride sınıflandırılır: denetimli öğrenme, denetimsiz öğrenme ve pekiştirmeli öğrenme. Bu algoritmalar, verilerden içgörüler elde etmek, sınıflandırmalar yapmak ve tahminlerde bulunmak için istatistiksel yöntemler kullanır. Günümüzde, birçok sektördeki büyük veri projelerinde, daha doğru sonuçlar elde etmek ve iş süreçlerini optimize etmek için bu algoritmalar yoğun bir şekilde kullanılmaktadır.

Makine öğrenimi algoritmalarının uygulama alanları, tedarik zinciri yönetiminden, lojistik ve perakendeye kadar geniş bir yelpazeye sahiptir. Bu algoritmalar, görevlerin otomatikleştirilmesi, verimliliğin artırılması ve daha kişiselleştirilmiş deneyimler sağlanması açısından önemli bir rol oynamaktadır.



**Şekil 8: Makine Öğrenmesi Türleri ve Algoritmalar**



**Şekil 9: Makine Öğrenmesi Türleri ve Algoritmaları Gösterimi**

## Makine Öğrenmesi Algoritması Seçerken Göz Önünde Bulundurulması Gereken Faktörler

Makine öğrenmesi algoritmasını seçerken dikkate almanız gereken birkaç temel faktör bulunmaktadır. Bu faktörler, sorunun doğasına, verinizin özelliklerine ve emrinizdeki kaynaklara göre değişkenlik gösterebilir. İşte bunlar:

### 1. Veri Türü

İlk adım, sahip olduğunuz verinin türünü belirlemektir. Bu, hangi öğrenme yöntemini kullanmanız gerektiğini doğrudan etkiler:

* ***Etiketli Veri (Gözetimli Öğrenme):*** Eğer verinizde her örneğe karşılık bir etiket (sonuç) varsa, gözetimli öğrenme yöntemleri kullanabilirsiniz. Bu durumda, denetimli algoritmalar (örneğin, Lojistik Regresyon, Karar Ağaçları) etkili olacaktır.
* ***Etiketsiz Veri (Gözetimsiz Öğrenme):*** Eğer verinizde etiketler yoksa, verinin gizli yapılarını keşfetmek için gözetimsiz öğrenme yöntemleri (örneğin, Kümeleme, PCA) uygundur.
* ***Etkileşimli Öğrenme (Pekiştirmeli Öğrenme):*** Eğer modelin, çevresiyle etkileşime girerek öğrenmesi gerekiyorsa, pekiştirmeli öğrenme yöntemleri (örneğin, Q-Learning, PPO) tercih edilebilir.

### 2. Problemin Karmaşıklığı

Çözmeye çalıştığınız problemin karmaşıklığı, algoritma seçimini etkileyen bir diğer önemli faktördür:

1. ***Karmaşık Problemler:*** Eğer daha karmaşık bir yapıya sahip, doğrusal olmayan ve yüksek boyutlu verilerle çalışıyorsanız, derin öğrenme algoritmaları veya topluluk yöntemleri (örneğin, Random Forest, Gradient Boosting) daha iyi sonuçlar verebilir. Ancak, bu tür yöntemler daha fazla hesaplama gücü ve veri gerektirir.

### 3. Hesaplama Kaynakları

Emrinizdeki hesaplama kaynakları, algoritma seçiminde önemli bir rol oynar:

* ***Sınırlı Kaynaklar:***Eğer işlemci gücünüz sınırlıysa, daha basit algoritmalar (örneğin, Lojistik Regresyon, KNN) daha uygun olabilir, çünkü bu algoritmalar daha az kaynak tüketir.
* ***Gelişmiş Kaynaklar:*** Eğer güçlü donanım (GPU, TPU vb.) kaynaklarınız varsa, derin öğrenme modelleri (örneğin, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs)) gibi kaynak yoğun algoritmalar tercih edilebilir.

### 4. Yorumlanabilirlik ve Doğruluk

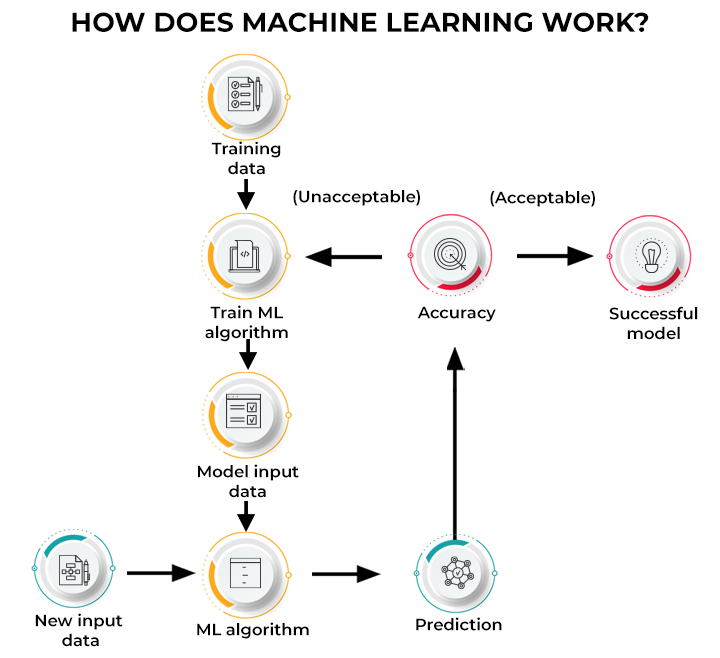
Algoritmanın doğruluğu ve yorumlanabilirliği arasında bir denge kurmanız gerekebilir:

* ***Doğruluk:*** Ancak, daha karmaşık ve yüksek doğruluğa sahip modeller kullanmanız gerekiyorsa, sinir ağları ve ensemble yöntemleri (örneğin, XGBoost, LightGBM) gibi daha güçlü ama açıklaması zor algoritmalar tercih edilebilir.

## Makine Öğrenimi Algoritmaları Nasıl Çalışır?

Makine öğrenimi algoritmalarının nasıl çalıştığını anlamak için, bu algoritmaların üç ana bileşene dayandığını söyleyebiliriz:

**Şekil 10: Makine Öğrenmesi Nasıl Çalışır?**



### Karar Süreci:

Makine öğrenimi algoritmaları, genellikle bir tahmin veya sınıflandırma yapmak için kullanılır. Bu süreç, algoritmanın, etiketlenmiş veya etiketlenmemiş veri üzerinde çalışarak, verilerdeki kalıpları öğrenmesini sağlar. Algoritma, verilen giriş verilerine dayanarak, veri setindeki bir model hakkında bir tahminde bulunur.

**Hata Fonksiyonu:**

Hata fonksiyonu, modelin tahminlerini değerlendirir. Eğer eğitim veri kümesinde bilinen örnekler mevcutsa, hata fonksiyonu, modelin tahminini doğru yanıtlarla karşılaştırarak doğruluğu ölçer. Bu karşılaştırma, modelin hatalarını anlamasına ve iyileştirilmesine yardımcı olur.

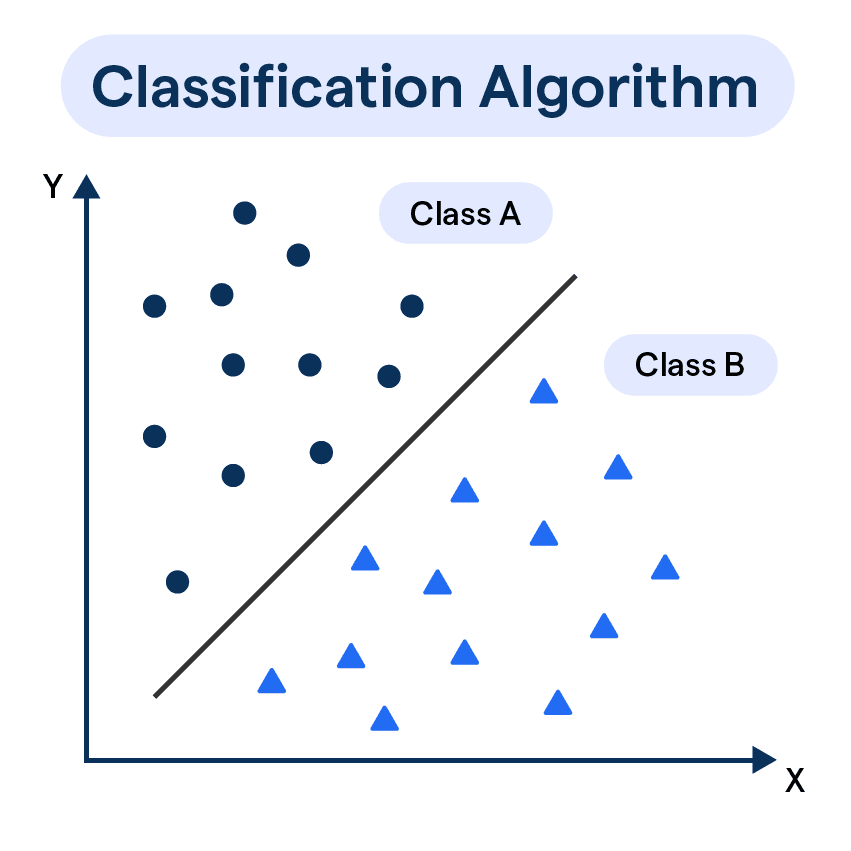
### Model Optimizasyonu Süreci:

Model, eğitim verilerine uyum sağlamak için optimize edilir. Tahminler ile doğru örnekler arasındaki tutarsızlıklar en aza indirilir ve ağırlıklar güncellenerek hata oranı azaltılır.

Denetimli öğrenmede, model girdiler ve doğru çıktılar arasındaki ilişkileri öğrenir. Kayıp fonksiyonu üzerinden hata ölçülür ve ağırlıklar optimize edilerek doğruluk artırılır.

## SINIFLANDIRMA ALGORİTMALARI

Sınıflandırma algoritmaları, verileri belirli kategorilere doğru bir şekilde atamak için kullanılır. Bu algoritmalar, veri kümesindeki belirli varlıkları tanır ve bunları doğru etiketlerle sınıflandırmaya çalışır.



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

Destek Vektör Makineleri (SVM), sınıflar arasında maksimum marjini sağlayan hiper düzlemi belirleyerek veri noktalarını sınıflandıran güçlü bir makine öğrenimi algoritmasıdır. Yüksek boyutlu uzaylarda verimli çalışır ve doğrusal olmayan durumlarda çekirdek fonksiyonları kullanarak sınıflandırmayı gerçekleştirebilir. SVM, sınıflandırma ve regresyon problemlerinde kullanılabilir, ancak en çok sınıflandırma görevlerinde tercih edilir.

**Şekil 11: Sınıflandırma Algoritması Gösterimi**

1. **Avantajlar**

Yüksek Boyutlarda Etkili: Özellik sayısının fazla olduğu veri setlerinde bile etkili sonuçlar verir.

Çekirdek Fonksiyonları: Doğrusal olmayan problemlerde kullanılabilir. Polinom veya RBF gibi çekirdek fonksiyonlarıyla sınıflandırma yapılabilir.

Marjin Maksimizasyonu: Farklı sınıfları ayıran karar sınırını belirlerken, iki sınıf arasındaki marjini en üst düzeye çıkarır.

* **Dezavantajlar**

Büyük Veri Setleri: Büyük veri setlerinde hesaplama maliyeti yüksek olabilir.

Çekirdek ve Parametre Seçimi: Çekirdek fonksiyonu ve düzenleme parametresi (C) seçiminde zorluklar yaşanabilir.

Yorumlanabilirlik: Modelin yorumlanması diğer algoritmalara göre daha zordur.

* **Kütüphaneler**

Python'da SVM'ler için yaygın olarak kullanılan kütüphane:

Scikit-Learn: SVC sınıfı ile SVM uygulanabilir.

* **Modeller**

Hiper Düzlem: Karar sınırını belirleyen düzlem; sınıflar arasındaki mesafeyi maksimize eder.

Destek Vektörleri: Hiper düzleme en yakın olan veri noktalarıdır ve karar sınırını belirlemede kritik rol oynar.

***Çekirdek Fonksiyonları:***

*Lineer Çekirdek:* Doğrusal sınıflandırma için uygundur.

*RBF (Radial Basis Function) Çekirdeği:* Doğrusal olmayan sınıflandırmalar için kullanılır.

*Polinom Çekirdeği:* Verileri polinom bir dönüşümle ayırır.

* **Uygulamalar**

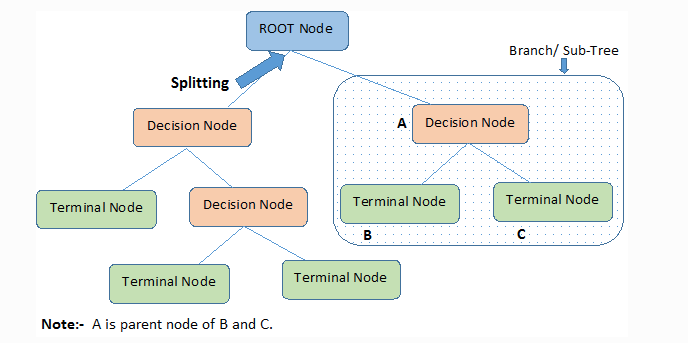
Görüntü Sınıflandırma: Görüntülerdeki nesneleri sınıflandırmak için kullanılır.

Metin Kategorizasyonu: Spam tespiti veya belge sınıflandırması gibi görevlerde tercih edilir.

Biyoenformatik: Gen ifadelerinin analizi ve protein sınıflandırması gibi problemleri çözmek için kullanılır.

### Karar Ağaçları (Decision Trees)

Karar ağaçları, verileri girdi özelliklerine göre bölerek hedef değişkeni tahmin etmek için kullanılan bir makine öğrenimi algoritmasıdır. Ağaç yapısında, her düğüm bir özelliği temsil ederken, dallar olası sonuçları ve yaprak düğümleri sınıf veya tahmin edilen değeri temsil eder.



**Şekil 12: Karar Ağaçları Gösterimi**

#### Avantajlar

Yorumlanabilirlik: Ağaç yapısı, sonuçların görselleştirilmesi ve yorumlanmasını kolaylaştırır.

Hızlı Modelleme: Veriler için hızlı ve verimli şekilde tahmin yapabilir.

* **Dezavantajlar**

Aşırı Uyum: Ağaçların derinleşmesi, modele aşırı uyuma neden olabilir.

Kararsızlık: Veri setindeki küçük değişiklikler farklı ağaç yapılarına yol açabilir.

Düşük Performans: Tek başına kullanıldığında karmaşık verilerde performansı düşük olabilir.

* **Kütüphaneler**

Karar ağaçlarını uygulamak için kullanılan bazı popüler Python kütüphaneleri:

XGBoost: Daha güçlü ve optimize edilmiş karar ağaçları modelleri sunar.

* **Modeller**

Karar ağaçları, farklı türlerde uygulanabilir:

Karar Ağacı Sınıflandırıcı (Decision Tree Classifier): Verileri sınıflara ayırır.

Karar Ağacı Regresyonu (Decision Tree Regressor):Sürekli değişkenleri tahmin eder.

Random Forest: Birden çok karar ağacının birleşimidir, daha güçlü tahminler yapar.

Gradient Boosting: Hataları minimize ederek model performansını artırır.

* **Uygulamalar**

Tıp: Hastalık teşhisi, ilaç etkisi analizi.

Finans: Kredi risk analizi, dolandırıcılık tespiti.

Pazarlama: Müşteri segmentasyonu, hedef kitlenin belirlenmesi.

Mühendislik: Sistem arızalarının tahmini, kalite kontrol.

### Sinir Ağları (Neural Networks)

Sinir ağları, makine öğreniminde kullanılan ve insan beyninin işleyişini taklit ederek verilerden öğrenme sağlayan güçlü algoritmalardır. Birbirine bağlı düğüm (nöron) katmanlarından oluşan bu modeller, özellikle karmaşık doğrusal olmayan ilişkilerin çözümünde etkilidir. Sinir ağları, genellikle büyük veri kümelerini işlemek ve yüksek doğruluklu tahminler yapmak için tercih edilir. Derin öğrenme algoritmalarının temel yapı taşıdır.

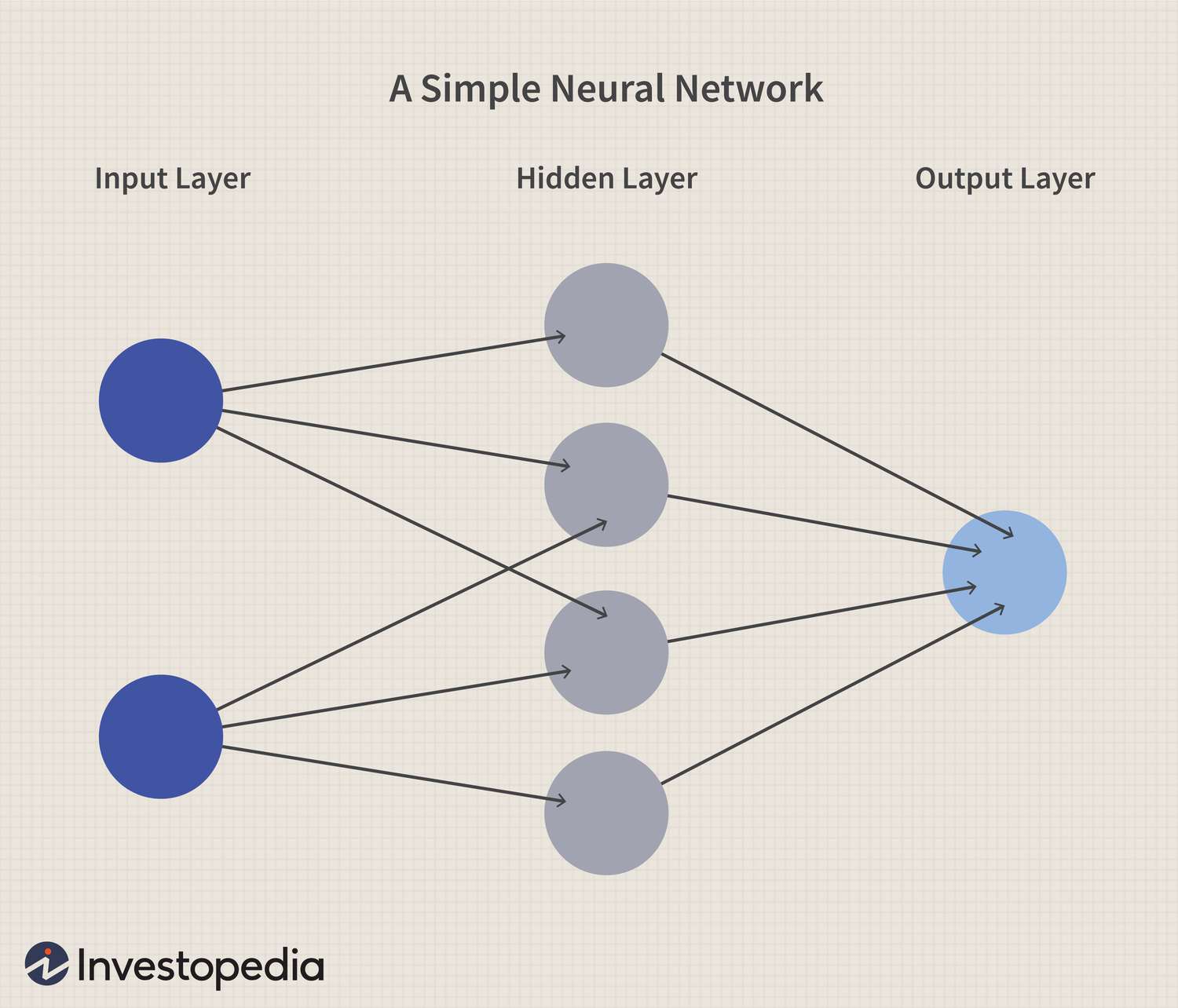
Sinir ağları, veri üzerinde tahmin ve öğrenme işlemlerini gerçekleştirirken girişlerden ağırlıklar, önyargı (bias), aktivasyon fonksiyonları ve çıktı değerleri gibi kavramlardan faydalanır. Bu mekanizma şu şekilde işler:

* Giriş katmanı, veriyi alır ve bir dizi işlem için gizli katmanlara iletir.
* Her bir düğüm, aldığı girdilere belirli ağırlıklar uygular ve sonuçları bir aktivasyon fonksiyonundan geçirerek çıktı üretir.
* Üretilen çıktı, bir sonraki katmana iletilir ve bu işlem son katmana kadar devam eder.
* Çıkış katmanı, problemin sınıflandırma veya regresyon gibi türüne göre son tahmini yapar.

Sinir ağları, genellikle hata fonksiyonunu minimize etmek amacıyla geri yayılım (backpropagation) ve gradyan iniş (gradient descent) yöntemlerini kullanır. Bu süreçte ağırlıklar ve önyargılar, hata oranını azaltacak şekilde güncellenir.

* **Avantajlar**

**Şekil 13: Sinir Ağları Gösterimi**



***Doğrusal Olmayan İlişkileri Öğrenme:*** Verilerdeki karmaşık desenleri ve ilişkileri anlamakta son derece başarılıdır.

***Esneklik:*** Çok çeşitli problem türlerinde (sınıflandırma, regresyon, kümeleme vb.) uygulanabilir.

***Özellik Çıkarma Yeteneği:*** Görsel, metinsel ve işitsel gibi farklı veri türlerinde, önemli özellikleri otomatik olarak öğrenebilir.

* **Dezavantajlar**

***Hesaplama Gücü Gereksinimi:*** Büyük miktarda veri işlemek için yüksek performanslı donanımlara ihtiyaç duyar (ör. GPU'lar).

***Aşırı Uyum (Overfitting):*** Veri miktarı yetersiz olduğunda veya model çok karmaşık olduğunda, eğitim verisine fazla uyum sağlama eğilimi gösterebilir.

* **Algoritma Türleri ve Uygulamaları**

***Sinir Ağları Türleri***

**Yapay Sinir Ağları (Artificial Neural Networks - ANN):**  
Geleneksel çok katmanlı yapılardır. Sınıflandırma ve regresyon gibi görevlerde kullanılır.

**Derin Sinir Ağları (Deep Neural Networks - DNN):**  
Çok sayıda gizli katmandan oluşur ve genellikle büyük veri setleriyle çalışır.

**Tekrarlayan Sinir Ağları (Recurrent Neural Networks - RNN):**  
Zaman serileri ve sekans verilerinde, bir önceki zaman adımındaki bilgiyi işler.

**Evrişimli Sinir Ağları (Convolutional Neural Networks - CNN):**  
Görüntü işleme ve görsel tanıma görevlerinde kullanılır.

**Güçlendirilmiş Sinir Ağları (Reinforced Neural Networks):**  
Pekiştirmeli öğrenme görevlerinde tercih edilir.

* **Uygulamalar**

Sinir ağları, birçok endüstride farklı uygulamalarda yaygın olarak kullanılır:

***Enerji Tahmini:*** Enerji tüketimini ve talep modellerini tahmin etmede kullanılır.

***Algoritmik Ticaret:*** Finansal piyasalardaki karmaşık fiyat hareketlerini analiz ederek ticaret kararları alır.

***Hava Durumu Tahmini:*** Atmosferik verilerden hava tahmini yapar.

***Görüntü İşleme:*** Nesne tanıma, yüz tanıma ve görüntü segmentasyonu gibi görevlerde uygulanır.

***Doğal Dil İşleme (NLP):*** Metin sınıflandırma, dil modeli oluşturma ve çeviri sistemleri gibi görevlerde kullanılır.

***Konuşma Tanıma:*** Sesli komutların tanınması ve işlenmesinde etkili sonuçlar sağlar.

* **Sinir Ağlarının Çalışma Prensibi**

Sinir ağlarının temeli, insan beynindeki nöronların çalışma prensibine dayanır. Her nöron (düğüm):

* Kendisine iletilen girdileri alır.
* Girdi değerlerini, ağırlıklar ile çarpar ve bir önyargı ekler.
* Aktivasyon fonksiyonu ile çıktı üretir.
* Bu çıktıyı bir sonraki katmana iletir.

Modelin eğitimi sırasında:

* ***İleri Yayılım (Forward Propagation*):** Veri, girişten başlayarak çıktı katmanına kadar işlenir.
* ***Geri Yayılım (Backpropagation*):** Hata oranı hesaplanır ve ağırlıklar, hatayı azaltacak şekilde güncellenir.

Sinir ağları, kayıp fonksiyonunu minimize ederek öğrenir. Kayıp fonksiyonu sıfıra yaklaştıkça modelin doğruluğu artar.

### K-En Yakın Komşular (k-NN)

K-En Yakın Komşular (k-NN) algoritması, bir veri noktası için sınıflandırma veya regresyon yaparken, en yakın k komşusunun etiketlerine veya değerlerine dayalı bir tahminde bulunur. Yani, bir örneği, özelliklerine göre en yakın k komşusunun çoğunluk oyu alacak şekilde sınıflandırır. Bu algoritma, parametrik olmayan ve tembel bir öğrenme algoritmasıdır, çünkü eğitim aşaması yoktur ve tüm eğitim verisi, tahmin yapılırken kullanılır.



**Şekil 14: K-NN Algoritması Örnek Gösterimi**

* **Avantajlar**

***Basit ve Etkili:*** K-NN, anlaşılması ve uygulanması kolay bir algoritmadır. Az sayıda parametre ile hızlı bir şekilde çözüm sunabilir.

***Parametrik Olmayan:*** Model, veri setine bağlı olarak doğrudan öğrenir ve herhangi bir parametreyi önceden tahmin etmez.

***Esneklik:*** Hem sınıflandırma hem de regresyon problemlerine uygulanabilir.

***Uygulama Kolaylığı:*** Özellikle küçük veri kümelerinde hızlıca sonuç verebilir.

* **Dezavantajlar**

***Hesaplama Maliyetli:*** K-NN, tüm veri kümesindeki her bir nokta ile mesafeleri hesapladığı için büyük veri setlerinde hesaplama maliyeti çok yüksektir.

***Öğrenme Süreci Yoktur:*** Eğitim aşaması olmadığından, her tahmin yapıldığında tüm veri kümesi üzerinden işlem yapılması gerekir.

***Veri Normalizasyonu*:** Özellikler arasındaki farklı ölçekler, K-NN'nin performansını etkileyebilir. Değişkenler normalleştirilmezse, daha yüksek aralıklı özellikler, algoritmayı önyargılı hale getirebilir.

***Bellek Kullanımı:*** Tüm eğitim verisinin saklanması gerektiği için büyük veri kümeleriyle çalışırken bellek tüketimi yüksektir.

* **Kullanılan Modeller ve Yöntemler**

K-NN, aşağıdaki yöntemlerle veri noktalarının etiketlerini tahmin eder:

***Sınıflandırma:*** Veri noktasının sınıfı, en yakın k komşusunun sınıf etiketlerinin çoğunluğuna dayanarak belirlenir. Örneğin, eğer 5 komşudan 3’ü sınıf A, 2’si sınıf B ise, yeni örnek sınıf A'ya atanır.

***Regresyon:*** Veri noktası için, en yakın k komşusunun değerlerinin ortalaması alınarak bir tahmin yapılır. Örneğin, konut fiyatı tahminlerinde, benzer konutların fiyatlarının ortalaması yeni konutun tahmin fiyatını verebilir.

* **Uygulama Alanları**

K-NN algoritması, çeşitli alanlarda yaygın olarak kullanılmaktadır:

***Öneri Sistemleri:*** Kullanıcılar benzer tercihlere sahip diğer kullanıcılarla karşılaştırılarak öneriler yapılabilir.

***Örüntü Tanıma:*** Görüntüler veya ses örüntüleri tanımlamak için kullanılabilir.

***Anomali Belirleme:*** Verinin normal yapısından farklı olan anormal durumları tespit etmek için k-NN kullanılabilir.

***Doküman Sınıflandırma*:** Metinlerin benzerliğine göre dokümanları kategorilere ayırabilir.

* **Kullanılan Kütüphaneler**

K-NN algoritması, Python'da çeşitli kütüphanelerle uygulanabilir:

***Scikit-learn:*** k-NN sınıflandırma ve regresyon işlemleri için KNeighborsClassifier ve KNeighborsRegressor sınıfları sağlar.

***TensorFlow / Keras:*** Derin öğrenme modellerine entegre edilen k-NN tabanlı modeller için kullanılabilir.

* **K-NN İçin Dikkat Edilmesi Gerekenler**

***K Değeri Seçimi:*** k değeri (komşu sayısı), doğruluğu doğrudan etkiler. Çok küçük bir k değeri, gürültüye duyarlı olabilirken, çok büyük bir k değeri modelin genelleştirme kapasitesini zayıflatabilir.

***Mesafe Ölçümü:*** En yaygın mesafe ölçümü Öklid mesafesidir, ancak farklı türdeki veriler için Manhattan mesafesi, Minkowski mesafesi veya Mahalanobis mesafesi de kullanılabilir.

***Veri Normalizasyonu*:** Özelliklerin farklı ölçekte olması k-NN’nin doğruluğunu etkiler. Bu yüzden veriler normalleştirilmelidir (örneğin, Min-Max scaling veya Z-score normalization).

### Lojistik Regresyon

Lojistik regresyon, bağımlı değişkenin kategorik olduğu durumlarda, özellikle ikili sınıflandırma problemlerinde (örneğin, evet/hayır, doğru/yanlış) kullanılan bir istatistiksel modeldir. Model, bağımsız değişkenlerin doğrusal bir kombinasyonu ile tahmin edilen logaritmik olasılıkları temel alarak, her bir örneğin belirli bir kategoriye ait olma olasılığını tahmin etmek için lojistik (sigmoid) fonksiyonu kullanır.

* **Avantajlar**

***Basit ve Etkili:*** Kullanımı kolaydır ve genellikle başlangıç seviyesinde bir sınıflandırma yöntemi olarak tercih edilir.

***Doğrusal Varsayım:*** Model, bağımsız değişkenler ile logaritmik olasılıklar arasında doğrusal bir ilişki olduğunu varsayar.

***Olasılık Tahmini:*** Model, her bir örneğin belirli bir sınıfa ait olma olasılığını üretir. Bu olasılıklar, genellikle 0,5 eşik değeri ile sınıflandırma yapmak için kullanılır.

* **Dezavantajlar**

***Doğrusal İlişki Varsayımı:*** Bağımsız değişkenler ile sonuçlar arasında doğrusal olmayan bir ilişki varsa, performansı düşer.

***Büyük Veri Setleri****:* Karmaşık veri setlerinde yetersiz kalabilir.

**Çoklu Sınıf Durumları**: Çıkışların ikiden fazla sınıf içerdiği durumlar için genişletilmesi gerekir (örneğin, multinominal lojistik regresyon).

* **Kütüphaneler**

Python'da lojistik regresyon için en yaygın kullanılan kütüphane:

***Scikit-Learn:*** LogisticRegression sınıfı ile lojistik regresyon uygulanabilir.

* **Modeller**

Log-Odds Modeli: Bağımlı değişkenin olasılıklarını logaritmik olarak ifade eden Logit fonksiyonu kullanılır:

Logit(p) = ln = +



***Sigmoid Fonksiyonu:*** Log-odds değerlerini olasılık tahminlerine dönüştürür:



* **Uygulamalar**

***E-posta Spam Tespiti****:* Bir e-postanın spam olup olmadığını tahmin etmek.

***Hastalık Teşhisi****:* Hastanın belirli bir hastalığa yakalanma olasılığını belirlemek.

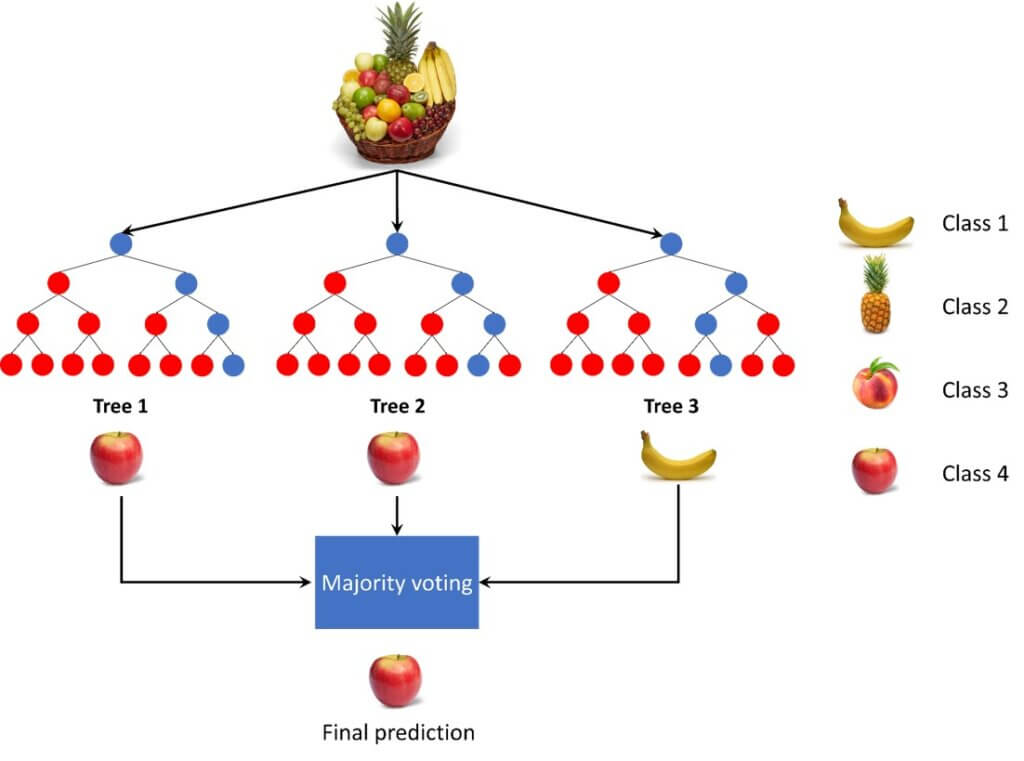
***Kredi Puanlama****:* Bir bireyin kredi riskini değerlendirmek.

***Pazarlama****:* Bir müşterinin bir ürünü satın alıp almayacağını tahmin etmek.

Lojistik regresyon, ikili sınıflandırma problemlerine hızlı, basit ve etkili bir çözüm sunar. Ancak, daha karmaşık problemler için genişletilmiş modeller veya alternatif yaklaşımlar gerekebilir.

### Rastgele Orman (Random Forest)

Rastgele Orman, farklı veri alt kümeleri ve özelliklerle eğitilmiş birden çok karar ağacının birleşiminden oluşan bir makine öğrenimi yöntemidir. Topluluk yöntemi sayesinde model doğruluğu artırılır ve aşırı uyum (overfitting) kontrol edilir.  
Her ağaç bağımsız olarak eğitilir ve sınıflandırma için "oylama", regresyon için ise "ortalama" alınır.



**Şekil 15: Rastgele Orman Algoritması Gösterimi**

* **Avantajlar**

***Aşırı Uyumun Azaltılması:*** Bireysel karar ağaçlarına kıyasla aşırı öğrenmeyi kontrol eder.

**Esneklik:** Hem sınıflandırma hem de regresyon problemlerinde kullanılabilir.

***Büyük Veriyle Çalışma:*** Yüksek boyutlu ve karmaşık veri kümelerinde iyi performans gösterir.

***Kararlılık:*** Veri setindeki küçük değişikliklerden daha az etkilenir.

* **Dezavantajlar**

***Hesaplama Maliyeti:*** Daha fazla işlem gücü ve kaynak gerektirir.

***Yorumlanabilirlik:*** Bireysel karar ağaçlarına kıyasla daha az şeffaf olabilir.

* **Kütüphaneler**

***Scikit-Learn:*** Kullanımı kolay temel bir Rastgele Orman implementasyonu sunar.

***XGBoost ve LightGBM:*** Büyük veri setleri için optimize edilmiş daha hızlı ve güçlü modeller sağlar.

***H2O.ai:*** Büyük ölçekli veri analizleri için etkili bir platformdur.

* **Uygulamalar**

***Finans*:** Kredi risk analizi, finansal tahminler.

***Sağlık:*** Hastalık teşhisi, ilaç etkisi analizi.

***Pazarlama:*** Müşteri segmentasyonu, hedef kitle analizi.

***Mühendislik:*** Sistem arızalarının tahmini, kalite kontrol.

### Gradient Boosting ve AdaBoosting Algoritmaları

Gradient Boosting ve AdaBoosting, her biri zayıf öğrenicileri birleştirerek güçlü tahmin modelleri oluşturmayı amaçlayan "boosting" algoritmalarına örneklerdir. Bu algoritmalar, doğruluk oranını artırmak ve hata düzeylerini minimize etmek için ardışık modellemeye dayalıdır. Temelde, her iki algoritma da hatalı tahminlerde bulunan önceki modellerin hatalarını düzeltmek için sırasıyla yeni modeller oluşturur. Boosting, birkaç zayıf tahmincinin (doğruyu tahmin etme kapasitesi düşük modeller) birleştirilmesiyle güçlü ve daha doğru tahminciler elde edilmesini sağlar.

* **Gradient Boosting Algoritması**

Gradient Boosting, ilk tahminci modelden başlayıp, her yeni modelin, önceki modelin hatalarını düzeltmeye yönelik olarak oluşturulmasını sağlar. Bu algoritma, her iterasyonda modelin doğruluğunu artırarak tahminleri daha iyi hale getirir. Örneğin, **XGBoost**, **LightGBM** ve **CatBoost** gibi popüler varyasyonlar, bu tekniğin hızlı ve verimli bir şekilde uygulanmasını sağlar. Bu algoritmalar, özellikle büyük veri setlerinde yüksek doğrulukla sonuçlar üretir.

* **AdaBoosting Algoritması**

AdaBoost (Adaptive Boosting), önceki modelin hatalarını düzelten bir başka boosting algoritmasıdır. Ancak, Gradient Boosting'den farklı olarak, AdaBoost, her iterasyonda önceki modelde yanlış sınıflandırılan örneklere daha fazla ağırlık vererek yeni modeller oluşturur. AdaBoost'un temel özelliği, zayıf tahmincileri birleştirerek güçlü bir model oluşturmasıdır. Bu yaklaşımda, doğrusal olmayan ve karmaşık ilişkiler daha iyi yakalanabilir.

* **Avantajlar**

***Yüksek Doğruluk:*** Her iki algoritma da doğru sonuçlar için optimize edilmiş, güçlü modeller oluşturur.

***Çeşitli Veri Türlerini İşleyebilir:*** Hem sayısal hem de kategorik verilerle çalışabilir, çok farklı veri türlerine uyum sağlar.

***Esneklik:*** Hem regresyon hem de sınıflandırma problemleri için uygulanabilir.

***Model İyileştirmesi:*** Önceki modelin hatalarını düzeltme yeteneği, modellerin doğruluğunu artırır.

* **Dezavantajlar**

***Aşırı Uyum Riski:*** Yetersiz model parametre ayarlamaları veya veri temizliği yapılmadığında, bu algoritmalar aşırı uyuma (overfitting) eğilimli olabilir.

***Hesaplama Zorluğu:*** Büyük veri kümeleriyle çalışırken çok sayıda iterasyon ve hesaplama gerektirir, bu da yüksek işlem gücü ve zaman tüketebilir.

***Model Parametre Ayarları:*** İyi sonuçlar elde etmek için algoritmaların doğru parametrelerle ayarlanması gerekir. Bu, modelin başarısını doğrudan etkiler.

* **Kullanılan Modeller ve Yöntemler**

***Gradient Boosting Regresyonu ve Sınıflandırması:*** *Her* iki algoritma, sırasıyla yeni modeller ekleyerek hataları düzeltir. Modelin doğruluğu artırılır, ancak dikkat edilmesi gereken şey uygun hiperparametre ayarlarıdır.

***XGBoost:*** Bu, Gradient Boosting'in bir çeşidi olup, özellikle büyük veri kümelerinde hız ve doğruluk konusunda mükemmel sonuçlar verir. XGBoost, optimizasyon için gradyan tabanlı teknikleri kullanır ve genel olarak hız ve verimlilik açısından oldukça etkilidir.

***LightGBM:*** XGBoost'a benzer, ancak daha hızlı ve hafif olmasıyla bilinir. Çeşitli veri türleriyle verimli çalışır ve özellikle çok büyük veri kümeleri üzerinde etkili olabilir.

***CatBoost:*** Hem sayısal hem de kategorik verilerle daha iyi uyum sağlayan, kolayca optimize edilebilen bir başka gradient boosting varyasyonudur.

* **Uygulamalar**

***Web Arama Sıralaması:*** İnternet üzerindeki içerikleri daha etkili bir şekilde sıralamak için kullanılabilir.

***Müşteri Kaybı Tahmini:*** Bir müşterinin ürün veya hizmeti kullanmayı bırakma olasılığını tahmin etmek için kullanılabilir.

***Sigorta Riski Tahmini:*** Sigorta şirketlerinin, poliçe sahiplerinin gelecekteki risklerini tahmin etmesine yardımcı olur.

***Konut Fiyat Tahmini:*** Konut fiyatlarının tahmin edilmesinde oldukça etkili olabilir.

* **Kütüphaneler ve Araçlar**

***Scikit-learn:*** Python'da GradientBoostingClassifier ve GradientBoostingRegressor sınıflarını içerir. Bu sınıflar, gradient boosting algoritmalarını hızlı bir şekilde uygulamayı sağlar.

***XGBoost:*** XGBoost'un Python ve R sürümleri, büyük veri setlerinde yüksek hız ve doğruluk sağlamak için yaygın olarak kullanılmaktadır.

***LightGBM:*** Bu kütüphane, özellikle büyük veri kümeleriyle hızlı ve verimli çalışır.

***CatBoost:*** CatBoost, kategorik verilerle daha iyi uyum sağlayan ve kullanıcı dostu bir kütüphanedir.

* **Önemli Notlar**

***Hiperparametre Ayarları:*** Bu algoritmaların başarısı, kullanılan hiperparametrelerin doğru şekilde ayarlanmasına bağlıdır. Özellikle öğrenme oranı, ağaç derinliği ve veri setinin büyüklüğü gibi parametreler dikkatlice seçilmelidir.

***Aşırı Uyumdan Kaçınma:*** Bu algoritmalar, düzgün ayarlanmadığında aşırı uyuma (overfitting) eğilimlidir. Modelin doğru parametrelerle optimize edilmesi, doğruluğun korunması için çok önemlidir.

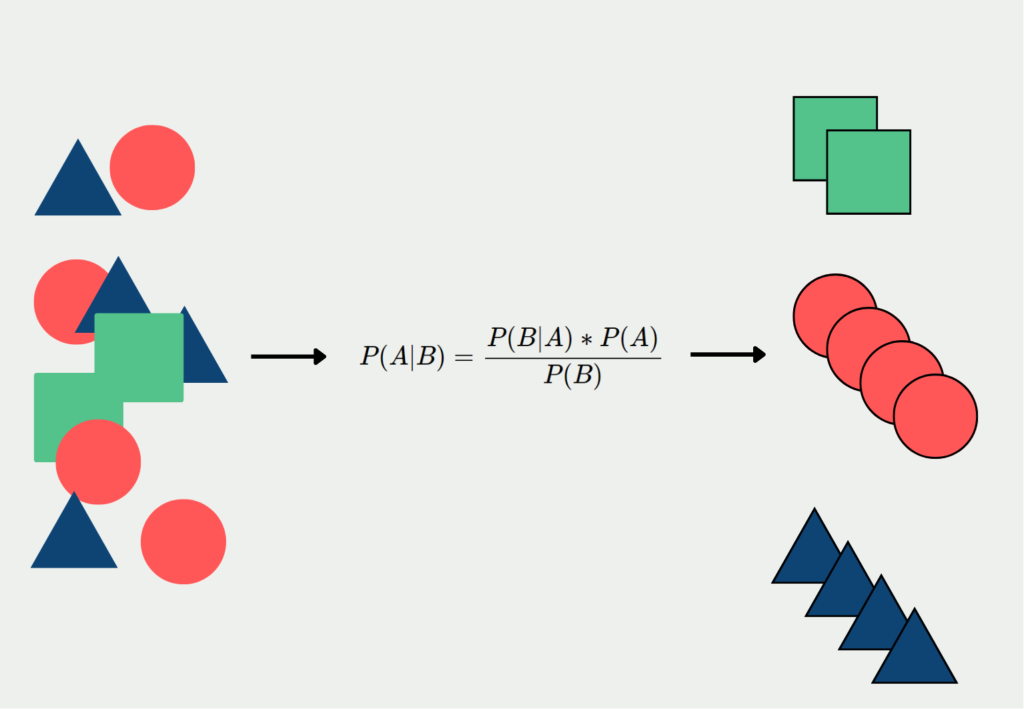
Gradient Boosting ve AdaBoosting algoritmaları, güçlü ve doğru tahmin modelleri oluşturmak için kullanılan etkili topluluk öğrenme teknikleridir. Her iki algoritma da, doğru parametrelerle kullanıldığında mükemmel sonuçlar verir ve makine öğrenimi uygulamaları için önemli araçlardır.

### Naive Bayes Algoritması

Naive Bayes, istatistiksel bir sınıflandırma algoritmasıdır ve Bayes Teoremi’ne dayalı çalışır. Bu algoritma, her özelliğin (değişkenin) birbirinden bağımsız olduğu varsayımıyla çalışır, bu yüzden "naive" (naif) olarak adlandırılır. Özellikler arasında bağımsızlık varsayımı, algoritmanın daha hızlı çalışmasını sağlar, ancak bazı durumlarda bu varsayım gerçek dünyada geçerli olmayabilir.

Naive Bayes, sınıflandırma problemlerinde yaygın olarak kullanılan basit fakat etkili bir tekniktir. Özellikle metin sınıflandırma, e-posta spam tespiti gibi uygulamalarda oldukça başarılıdır.

**Şekil 16: Naive Bayes Algoritması Gösterimi**



* **Avantajlar**

***Hızlı Eğitim ve Tahmin:*** Naive Bayes algoritması, büyük veri kümelerinde bile hızlı bir şekilde eğitilebilir ve tahmin yapabilir.

***Kolay Uygulama:*** Özellikle büyük veri setleri ve metin verisi gibi durumlarda oldukça kolay uygulanabilir.

***Düşük Hesaplama Maliyeti:*** Özellikler arası bağımsızlık varsayımı sayesinde, modelin öğrenme süreci düşük hesaplama maliyetine sahiptir.

***Çoklu Sınıf Desteği:*** Naive Bayes, çok sınıflı (multi-class) sınıflandırma problemleri için uygundur.

* **Dezavantajlar**

***Bağımsızlık Varsayımı*:** Özelliklerin birbirinden bağımsız olduğu varsayımı çoğu zaman gerçek dünyada geçerli olmayabilir, bu da modelin doğruluğunu etkileyebilir.

***Küçük Veri Setlerinde Düşük Performans:*** Özellikle çok küçük veri setlerinde, modelin tahminleri genellikle doğru olmayabilir.

***Sınırlı Model Esnekliği:*** Özellikler arasındaki karmaşık ilişkileri yakalamada zorluk yaşar, çünkü her özellik birbirinden bağımsız kabul edilir.

* **Kütüphaneler ve Modeller**

***Gaussian Naive Bayes:*** Sürekli özellikler için kullanılır ve özelliklerin normal dağıldığı varsayımına dayanır.

***Multinomial Naive Bayes:*** Genellikle metin sınıflandırması için kullanılır. Özellikler, belirli bir sınıfın olasılık dağılımına göre sayısal verilere dayanır.

* **Uygulamalar**

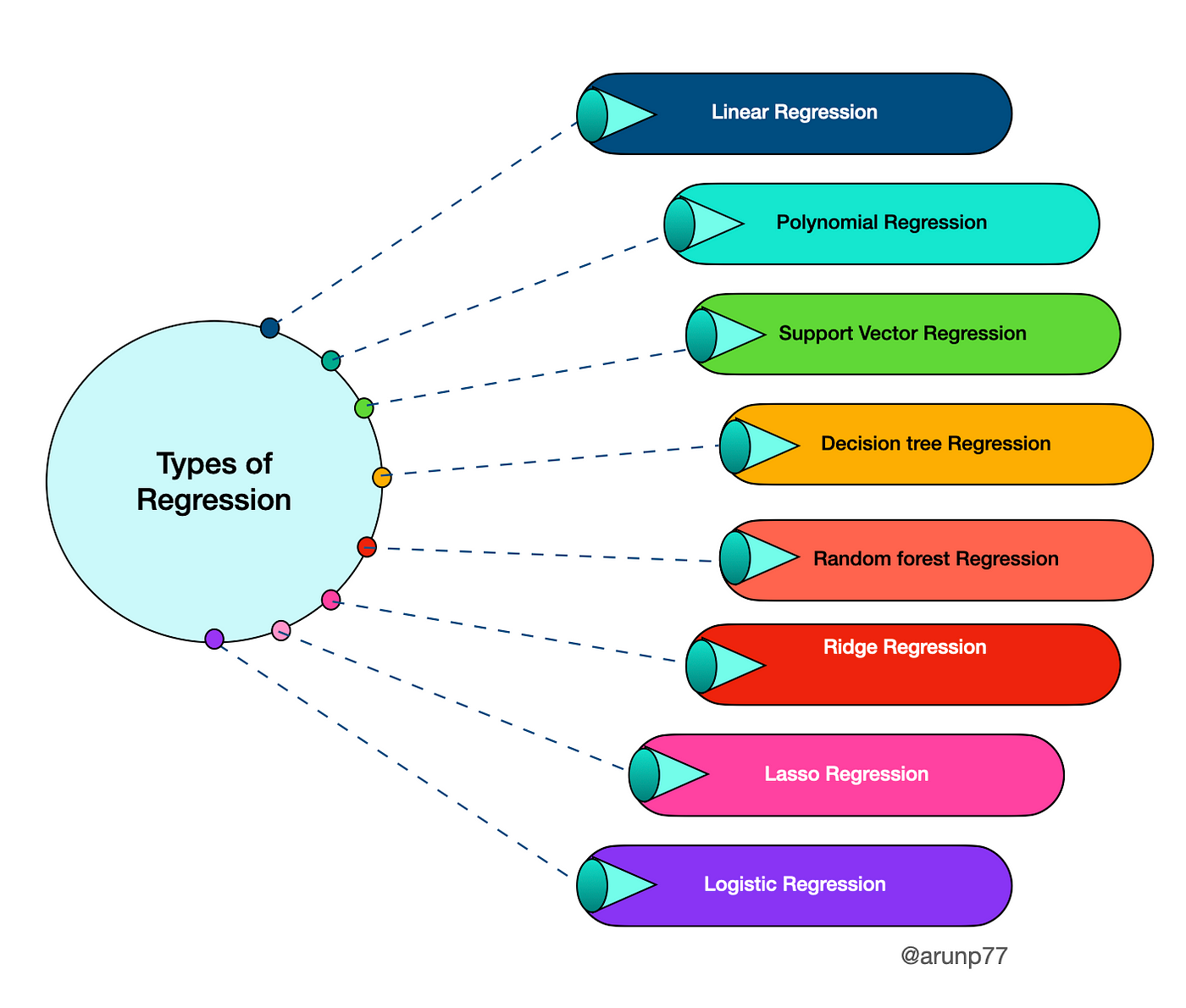
***E-posta Spam Filtreleme:*** E-posta mesajlarının spam olup olmadığını belirlemek için yaygın olarak kullanılır.

***Metin Sınıflandırma:*** Naive Bayes, haber başlıkları veya sosyal medya gönderileri gibi metin verilerini sınıflandırmak için etkili bir yöntemdir.

***Hangi Ürünlerin Satılacağına Dair Tahminler:*** *Perakende* sektöründe, belirli ürünlerin satış tahminlerinde kullanılabilir.

## REGRESYON ALGORİTMALARI

Regresyon algoritmaları, girdi özelliklerini kullanarak sürekli bir çıktı değişkeninin tahmin edilmesiyle ilgilidir. Emlak fiyatlarını, borsa trendlerini, müşteri kaybını veya satış tahminlerini öngörmek gibi çeşitli durumlarda kullanılabilir. Regresyon, bağımlı ve bağımsız değişkenler arasındaki ilişkiyi anlamak için yaygın olarak kullanılır. Doğrusal regresyon, lojistik regresyon ve polinom regresyon, popüler regresyon algoritmalarıdır.



**Şekil 17: Regresyon Algoritmaları**

### Doğrusal Regresyon

Doğrusal regresyon, bağımsız (girdi) ve bağımlı (çıktı) değişkenler arasındaki ilişkiyi bir doğrusal denklem ile modelleyen bir makine öğrenimi algoritmasıdır. Amaç, veri noktalarına en iyi şekilde uyan bir doğrusal regresyon doğrusu (Y = aX + b) oluşturarak tahmin yapmaktır.

* **Avantajlar**

***Basitlik:*** Modelin kurulumu ve anlaşılması kolaydır.

***Yorumlanabilirlik:*** Katsayılar, değişkenler arasındaki ilişkinin yönü ve büyüklüğünü açıkça gösterir.

***Hız:*** Küçük ve orta boyutlu veri setleri üzerinde hızlı bir şekilde çalışabilir

* **Dezavantajlar**

***Doğrusallık Varsayımı*:** Değişkenler arasında doğrusal bir ilişki olduğu varsayılır; bu durum, karmaşık ilişkilerde modelin yetersiz kalmasına neden olabilir.

***Aykırı Değerlere Hassasiyet:*** Model, veri setindeki uç değerlerden önemli ölçüde etkilenebilir.

***Çoklu Bağıntı Sorunu:*** Bağımsız değişkenler arasında yüksek derecede ilişki olduğunda, model performansı düşebilir.

* **Popüler Kütüphaneler**

***Scikit-Learn:*** Python'da doğrusal regresyon modelleri için kullanılan temel bir kütüphanedir.

***Statsmodels:*** İstatistiksel analiz ve doğrusal regresyon için detaylı çıktılar sağlar.

***TensorFlow ve PyTorch:*** Büyük ölçekli regresyon modellerini derin öğrenme çerçeveleri içinde oluşturabilir.

* **Uygulamalar**

***Konut Fiyat Tahmini:*** Evin büyüklüğü, oda sayısı gibi özelliklere dayanarak fiyat tahmini yapılabilir.

***Risk Yönetimi:*** Finans sektöründe risk seviyelerini tahmin etmek için kullanılır.

***Tıp*:** Yaş, boy, kilo gibi verilere dayanarak sağlık ölçütlerini tahmin etmek.

### Destek Vektör Regresyonu (SVR)

Destek Vektör Regresyonu (SVR), regresyon görevlerinde kullanılan bir tekniktir. SVR, verilen hedef değerler ile belirtilen bir marjdan daha büyük olmayan sapmalarla bir fonksiyon bularak, doğrusal olmayan veri setlerinde tahmin yapar. Bu, daha doğrusal olmayan ilişkileri modelleyebilme yeteneği sağlar.

* **Avantajlar**

***Yüksek boyutlu uzaylarda etkilidir****:* SVR, özellikle çok büyük veri setlerinde ve yüksek boyutlu özellik uzaylarında iyi performans gösterir.

***Aykırı değerlere karşı dayanıklıdır****:* SVR, aykırı verilere karşı daha dayanıklıdır çünkü yalnızca marj içindeki veriler kullanılarak model oluşturulur.

***Genelleme yeteneği güçlüdür****:* Aşağıdaki parametrelerin uygun şekilde seçilmesi durumunda aşırı uyum (overfitting) riski düşer.

* **Dezavantajlar**

***Çekirdek fonksiyonu ve düzenleme parametrelerine duyarlıdır****:* Modelin performansı, çekirdek fonksiyonu ve düzenleme parametrelerinin doğru seçimine bağlıdır. Bu parametrelerin yanlış ayarlanması modelin doğruluğunu etkileyebilir.

***Hesaplama maliyetli olabilir****:* Büyük veri setlerinde ve karmaşık hesaplamalarla çalışırken SVR'nin hesaplama süresi uzun olabilir.

***Modelin eğitilmesi zaman alabilir****:* SVR, büyük veri kümeleriyle çalışırken eğitim süresi anlamlı derecede artabilir.

* **Modeller ve Kütüphaneler**

***Scikit-learn (Python)****:* SVR sınıfı, Destek Vektör Regresyonu için yaygın olarak kullanılan bir modeldir. Kullanımı ve uygulanması oldukça basittir.

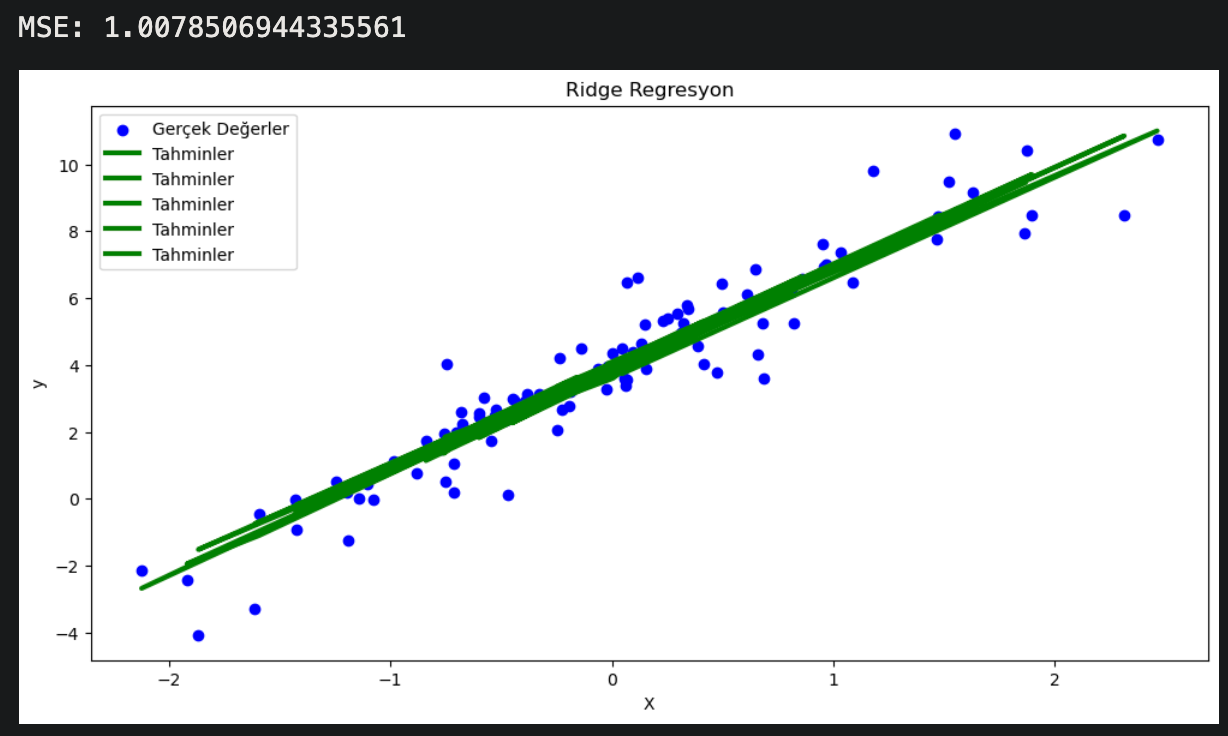
***LibSVM (C++/Python)****:* SVM'lerin temeli olan bu kütüphane, SVR uygulamaları için de uygundur.

* **Uygulama Alanları**

***Zaman serisi tahminleri****:* Ekonomik veriler, hava durumu tahminleri, üretim süreçlerinin optimizasyonu gibi uygulamalar.

### Ridge Regresyonu

Ridge regresyonu, doğrusal regresyona L2 düzenlemesi ekleyerek aşırı uyumu önler ve çoklu doğrusallık sorunlarını çözmeye yardımcı olur.



**Şekil 18: Ridge Regresyon**

* **Avantajları**

Aşırı uyumu azaltır.

Çoklu doğrusallık problemlerini iyi yönetir.

Tüm özelliklerin modelde yer almasını sağlar (katsayılar sıfıra indirgenmez).

* **Dezavantajları**

Tüm özellikler modelde kalır, bu nedenle gereksiz özellikler elenmez.

Düzenleme parametresi (lambda) dikkatli bir şekilde ayarlanmalıdır.

* **Kütüphaneler**

***Python:*** scikit-learn (Ridge sınıfı)

***R:*** glmnet

* **Modeller**

Ridge Regression (L2 Regularization)

* **Uygulama Alanları**

Ekonomik tahmin

Portföy optimizasyonu

### Lasso Regresyonu

Lasso regresyonu, bazı katsayıları sıfıra düşürerek özellik seçimini gerçekleştirmek için doğrusal regresyona L1 düzenlemesi ekler.

* **Avantajları**

Özellik seçimi yapar ve gereksiz değişkenleri sıfıra indirir.

Seyrek (sparse) modeller oluşturabilir.

* **Dezavantajları**

Düzenleme parametresinin (lambda) ayarlanması zordur.

Çoklu doğrusallık durumunda bazen yetersiz kalabilir.

* **Kütüphaneler**

***Python:*** scikit-learn (Lasso sınıfı)

***R:*** glmnet

* **Modeller**

Lasso Regression (L1 Regularization)

* **Uygulama Alanları**

Gen seçimi

Model seçimi

Finans

## KÜMELEME ALGORİTMALARI

Kümeleme algoritmaları, etiketlenmemiş verileri benzerliklerine göre gruplandırmak için kullanılır. Örneğin, meyve verileri üzerinde K-ortalama kümeleme gibi teknikler, meyveleri renk veya boyut gibi özelliklere göre kümelere ayırabilir. Hiyerarşik kümeleme, kümeler arasındaki ilişkileri inceleyerek bir hiyerarşi oluşturur. DBSCAN gibi yoğunluk tabanlı algoritmalar ise, veri eksikliği veya aykırı değerlerin bulunduğu durumlarda bile yoğun veri noktalarını bulabilir.

### Hiyerarşik Kümeleme

Hiyerarşik kümeleme, verileri hiyerarşik bir yapıya (ağaç yapısı) yerleştiren bir kümeleme algoritmasıdır. Bu yöntem, verileri benzerliklerine göre gruplandırarak bir küme ağacı oluşturur. Ağaç, her bir verinin küme ilişkisini gösterir. Hiyerarşik kümeleme, verilerin önceki kümelerle nasıl ilişkili olduğunu gösteren bir dendrogram (ağaç diyagramı) üretir.

* **Avantajlar**

***Önceden Tanımlı Küme Sayısı Gerektirmez*:** Kullanıcı, küme sayısını önceden belirlemek zorunda kalmaz. Küme sayısı hiyerarşi yapısı üzerinde analiz edilerek seçilebilir.

***Dendrogram ile Görselleştirme:*** Kümelerin nasıl birleştirildiği veya ayrıldığı görselleştirilebilir, bu da sonuçları daha anlaşılır kılar.

***Esneklik:*** Hem küçük hem de büyük veri setleri üzerinde etkili olabilir.

***İleriye Dönük Küme Sayısı Seçimi:*** Kullanıcı dendrogram üzerinde kesim yaparak istediği küme sayısını belirleyebilir.

* **Dezavantajlar**

***Hesaplama Maliyetleri:*** *Büyük* veri kümeleriyle çalışırken hesaplama açısından yoğun ve zaman alıcı olabilir.

***Ölçeklenebilirlik Sorunları:*** Veri boyutu arttıkça, algoritmanın performansı düşebilir.

***Büyük Veri Kümesinde Zorluklar:*** Çok büyük veri kümelerinde, bağlantı hesaplamaları ve dendrogram oluşturma işlemi çok fazla bellek ve işlem gücü gerektirebilir.

* **Kullanılan Modeller ve Bağlantı Yöntemleri**

Hiyerarşik kümeleme algoritmalarında kullanılan başlıca bağlantı yöntemleri şunlardır:

***Aglomeratif Yöntem (Bottom-Up)***

Verilerin her biri başlangıçta birer küme olarak kabul edilir.

Kümeler, en yakın iki küme birleştirilerek büyür.

***Bölen Yöntem (Top-Down)***

Tüm veriler başlangıçta tek bir küme oluşturur ve bu küme, daha küçük parçalara bölünür.

* **Uygulama Alanları**

Hiyerarşik kümeleme, aşağıdaki alanlarda yaygın olarak kullanılmaktadır:

***Sosyal Ağ Analizi:*** Sosyal medya verilerinde kullanıcılar arasında benzerlikler bulunarak gruplar oluşturulabilir.

***Genetik Veri Analizi:*** Gen dizilimlerinde benzer genleri gruplamak için kullanılabilir.

**Pazar Segmentasyonu:** Müşterileri benzer alışveriş alışkanlıklarına göre gruplandırmak için kullanılır.

* **Kullanılan Kütüphaneler**

Hiyerarşik kümeleme için popüler Python kütüphaneleri:

***Scikit-learn****:* Hiyerarşik kümeleme işlemleri için AgglomerativeClustering sınıfı sağlar.

***SciPy:*** scipy.cluster.hierarchy modülü ile hiyerarşik kümeleme yapılabilir ve dendrogramlar çizilebilir.

***NumPy:*** Verilerin işlenmesi ve hesaplama işlemleri için yaygın olarak kullanılır.

### DBSCAN (Gürültüye Sahip Uygulamaların Yoğunluk Tabanlı Mekansal Kümelenmesi)

DBSCAN (Density-Based Spatial Clustering of Applications with Noise), mesafe ve yoğunluk tabanlı bir kümelenme algoritmasıdır. Veriyi, belirli bir yoğunluk eşik değerine ve mesafeye göre gruplar. Bu algoritma, yoğun alanlarda bulunan noktaları küme olarak tanımlar ve düşük yoğunluklu alanlarda yer alan noktaları gürültü veya aykırı değer olarak kabul eder. Geleneksel geometrik yapılara bağlı kalmaksızın, rastgele şekilli küme yapılarını tanıyabilmesi, DBSCAN'ın önemli bir özelliğidir.

* **Avantajlar**

***Keyfi şekilli küme tanımlayabilme****:* Veriyi belirli geometrik şekillere (küre veya çizgi gibi) sınırlamadan, rastgele şekilli küme yapılarını tespit edebilir.

***Gürültüye dayanıklılık:*** Düşük yoğunluklu alanlarda yer alan aykırı değerleri (outliers) otomatik olarak ayırır ve analiz sırasında dikkate almaz.

***Parametre optimizasyonu ile esneklik****:* Çevresel yarıçap (ε) ve minimum nokta sayısı gibi yoğunluk parametreleri, çeşitli veri türlerine uyacak şekilde ayarlanabilir.

* **Dezavantajlar**

***Parametre hassasiyeti***: Algoritmanın performansı, ε (epsilon) ve minimum nokta sayısı parametrelerinin uygun bir şekilde ayarlanmasına bağlıdır. Yanlış ayarlar , ya fazla büyük ya da fazla küçük küme oluşumuna neden olabilir.

***Yüksek boyutlu verilerde performans kaybı****:* DBSCAN, düşük boyutlu verilere daha uygundur. Yüksek boyutlu veri setlerinde mesafeye dayalı hesaplamalar maliyetli olabilir.

***Kümelemeye uygun olmayan veri yapıları****:* Düzenli bir yoğunluk dağılımına sahip olmayan veri setlerinde, kümeleme performansı düşebilir.

* **Model**

DBSCAN modeli, şu özellikleriyle öne çıkar:

***Çekirdek Noktalar****:* Belirli bir yoğunluğun üzerinde bulunan noktalar.

***Sınır Noktaları****:* Bir kümenin kenarında yer alan, ancak kümenin bir parçası olarak kabul edilen noktalar.

***Gürültü Noktaları****:* Hiçbir kümenin parçası olmayan aykırı noktalar.

* **Kütüphaneler**

DBSCAN algoritmasını uygulamak için yaygın olarak kullanılan kütüphaneler ve dil desteği:

***Python****:* sklearn.cluster.DBSCAN

***R****:* dbscan paketi.

***Matlab****:* dbscan fonksiyonu.

***Julia****:* Clustering paketi.

* **Uygulama Alanları**

***Coğrafi Veri Analizi***: Coğrafi bilgi sistemlerinde (GIS) yoğunluk analizleri, şehir planlama, trafik izleme.

***Dolandırıcılık Tespiti****:* Finansal veri setlerindeki olağan dışı hareketlerin algılanması.

***Biyoloji****:* Genetik verilerde benzer genetik dizilimlerin belirlenmesi.

***Pazar Analizi****:* Tüketici davranışlarını gruplandırma ve şablon belirleme.

### Gauss Karışım Modelleri (GMM)

GMM, verilerin her biri bir kümeyi temsil eden birkaç Gauss dağılımının karışımından üretildiğini varsayar.

* **Avantajları**

Farklı şekil ve boyutlardaki kümeleri modelleyebilir.

Olasılıksal yumuşak atamalar ile esnek bir model sunar.

* **Dezavantajları**

Başlatmaya duyarlıdır ve yerel optimumlara takılabilir.

Büyük veri kümelerinde hesaplama maliyeti yüksek olabilir.

* **Kütüphaneler**

***Python:*** scikit-learn (GaussianMixture sınıfı)

* **Modeller**

Gaussian Mixture Model (GMM)

* **Uygulama Alanları**

Görüntü segmentasyonu

Anomali tespiti

### K-Means Algoritması

K-Means , makine öğreniminde kümeleme (clustering) problemlerini çözmek için kullanılan popüler ve basit bir algoritmadır. Denetimsiz öğrenme yöntemlerinden biridir ve verileri önceden tanımlanmamış gruplara ayırarak benzer özelliklere sahip veri noktalarını bir araya getirir. K-Means , özellikle kümeleme problemlerinde yaygın olarak tercih edilir.

* **Avantajlar**

***Basitlik ve Hız:*** Algoritma kolay anlaşılır ve büyük veri kümelerinde bile hızlı çalışır.

***Hesaplama Verimliliği:*** Büyük ölçekli veri kümelerinde etkili sonuçlar sağlar.

***Esneklik:*** Küme sayısı (k) seçildiğinde farklı veri türlerinde uygulanabilir.

* **Dezavantajlar**

***Küme Sayısının Belirlenmesi:*** K değerinin önceden belirlenmesi gerekir, bu da bazen zorluk yaratabilir.

***Hassasiyet:*** Başlangıç merkez noktalarına duyarlıdır, bu nedenle farklı sonuçlar elde edilebilir.

***Küme Şekilleri:*** Yalnızca küresel ve simetrik kümeleri etkili şekilde ayırabilir; karmaşık şekilli kümeler için uygun değildir.

* **Kütüphaneler**

Python'da K-Means algoritmasını uygulamak için yaygın olarak kullanılan kütüphaneler

***Scikit-Learn:*** KMeans sınıfı ile kolayca uygulanabilir.

***NumPy:*** Veri manipülasyonu ve küme merkezlerinin hesaplanması için kullanılabilir.

* **Modeller**

***Küme Merkezi:*** Her küme için ortalama bir nokta (centroid) belirler ve kümeyi bu merkeze yakın veri noktaları oluşturur.

***Öklid Mesafesi:*** Verileri kümelere ayırmak için veri noktalarının küme merkezine olan mesafesini ölçer.

* **K-Means Adımları**
* Küme sayısı k belirlenir.
* Veri kümesinde rastgele k adet küme merkezi (centroid) seçilir.
* Her veri noktası, en yakın küme merkezine atanır.
* Küme merkezleri, kümelere atanan noktaların ortalaması alınarak güncellenir.
* Adımlar 3 ve 4, kümeler sabitlenene veya belirlenen iterasyon sınırına ulaşılana kadar tekrarlanır.
* **Uygulamalar**

***Görüntü Sıkıştırma***: Piksel yoğunluklarına göre kümeler oluşturarak görüntülerin boyutunu küçültmek için kullanılır.

***Anomali Tespiti:*** Kümelerden uzak olan veri noktalarını anormal olarak tanımlamak için kullanılır.

## BOYUT AZALTMA ALGORİTMALARI

Boyutsallık azaltma, özellikle büyük ve karmaşık veri setlerinde analiz, görselleştirme ve modelleme süreçlerini daha verimli hale getirmek için kullanılan bir tekniktir. Bu yöntemler, veri kümesindeki özellik sayısını azaltarak yalnızca en anlamlı bilgileri korumayı amaçlar. Boyut azaltma , hem veri kaybını minimize etmek hem de hesaplama maliyetlerini düşürmek için kritik bir rol oynar.

### Temel Bileşen Analizi (PCA)

Temel Bileşen Analizi (PCA), bir veri kümesinin boyutunu azaltmak için kullanılan istatistiksel bir tekniktir. Verilerin maksimum varyansını yakalayarak yeni bir ortogonal özellikler kümesine (temel bileşenler) dönüştürür. PCA, orijinal özelliklerin doğrusal kombinasyonlarını oluşturarak, verinin karmaşıklığını azaltırken mümkün olduğunca fazla bilgi tutmayı amaçlar.

* **Avantajları**

***Boyut azaltma:*** Yüksek boyutlu verileri daha az bileşene indirerek analiz süresini ve hesaplama maliyetini azaltır.

***Görselleştirme*:** İki veya üç temel bileşenle yüksek boyutlu verilerin görselleştirilmesine olanak tanır.

***Gürültü azaltma:*** Verideki düşük varyanslı bileşenleri ortadan kaldırarak gürültüyü filtreler.

* **Dezavantajları**

***Doğrusal varsayım*:** PCA, yalnızca doğrusal ilişkileri yakalar; doğrusal olmayan ilişkilere duyarsızdır.

***Yorumlama zorluğu:*** Yeni bileşenlerin gerçek dünyadaki anlamını yorumlamak zor olabilir.

***Özellik ölçeklendirme:*** PCA'nın doğru çalışması için verilerin önceden standartlaştırılması (örneğin, z-skor normalizasyonu) gerekir.

***Bilgi kaybı:*** Bazı düşük varyanslı bileşenleri kaldırmak, bilgi kaybına yol açabilir.

* **Kütüphaneler**

***Python*:**

**scikit-learn:** PCA sınıfı

**numpy*:*** SVD tabanlı PCA implementasyonu

**matplotlib ve seaborn**: Görselleştirme için

***R:***

stats paketi (prcomp ve princomp fonksiyonları)

FactoMineR ve ggbiplot görselleştirme için

* **Modelleme**

***Standart PCA:*** Geleneksel doğrusal PCA yöntemidir ve tüm temel bileşenleri kullanır.

***Kernel PCA:*** Doğrusal olmayan ilişkileri yakalamak için çekirdek fonksiyonları kullanır.

***Incremental PCA:*** Büyük veri kümelerinde hesaplama verimliliği sağlamak için aşamalı öğrenme uygular.

***Sparse PCA:*** Seyrek temsiller oluşturmak için PCA'yı genişletir.

* **Uygulama Alanları**

***Veri Sıkıştırma:***

Yüksek boyutlu veri setlerini daha küçük boyutlara indirerek depolama ve iletim maliyetlerini azaltır.

***Gürültü Azaltma:***

Genetik veriler veya sinyaller gibi yüksek gürültülü veri kümelerinde kullanılabilir.

***Özellik Çıkarma:***

Görüntü işleme ve konuşma tanıma gibi alanlarda karmaşık veri kümelerinden anlamlı özellikler elde etmek için kullanılır.

***Finansal Modelleme:***

Varlık getirilerini analiz ederek temel faktörleri bulmada faydalıdır.

***Görselleştirme:***

İki boyutlu grafikleri kullanarak, müşteri segmentasyonu veya biyolojik verilerin keşfi gibi yüksek boyutlu veri kümelerini görselleştirir.

### T-Dağıtılmış Stokastik Komşu Gömme (T-SNE)

T-SNE, yüksek boyutlu verilerin görselleştirilmesi için kullanılan güçlü bir tekniktir. Verilerin yerel yapısını koruyarak boyutunu düşürür, böylece benzer veri noktaları birbirine yakın yerlerde gruplanır. T-SNE, özellikle karmaşık veri kümelerinin görselleştirilmesinde kullanılır ve bu sayede veri analizinin görsel bir şekilde anlaşılmasını sağlar.

* **Avantajları**

***Yerel yapı korunur:*** T-SNE, verilerin yerel benzerliklerini iyi bir şekilde korur ve benzer noktaları yakın tutar.

***Karmaşık veriler için etkili:*** Yüksek boyutlu, karmaşık veri kümelerinin görselleştirilmesi için kullanışlıdır.

* **Dezavantajları**

***Hesaplama maliyeti:*** Büyük veri kümelerinde hesaplama açısından oldukça yoğundur ve uzun süreler alabilir.

***Küresel yapıyı kaybeder:*** Verinin genel yapısını veya uzaklık ilişkilerini tam olarak koruyamaz.

***Parametre hassasiyeti*:** T-SNE'nin sonuçları, hiperparametre ayarlarına çok duyarlıdır.

* **Kütüphaneler**

***Python*:**

**scikit-learn:** T-SNE sınıfı

**bhtsne:** Büyük veri kümelemeleri için hızlı t-SNE algoritması

**matplotlib ve seaborn:** Sonuçları görselleştirmek için

***R:***

Rtsne paketi

**ggplot2:** Sonuçları görselleştirme için

* **Modelleme**

***Vanilla t-SNE:*** Temel t-SNE algoritmasıdır, yerel yapıyı korumada başarılıdır ancak büyük veri kümelerinde zorluk yaşar.

**Perplexity Tuning:** T-SNE'nin karmaşıklığını denetlemek için perplexity parametresi ayarlanabilir.

**Barnes-Hut t-SNE:** Büyük veri setlerinde hesaplama hızını artırmak için Barnes-Hut yaklaşımı kullanılır.

* **Uygulama Alanları**

***Küme Görselleştirme:***

Kümeler arasındaki farklılıkları görsel olarak anlamak, özellikle denetimsiz öğrenme algoritmalarında etkilidir.

***Yüksek Boyutlu Verileri Keşfetme:***

Görselleştirme ve analiz, genetik veri, metin verisi gibi karmaşık veri kümeleri için kullanılır.

***Anomali Algılama:***

Sınıflandırılmamış veri setlerinde anomali tespiti için yardımcı olabilir.

### Doğrusal Diskriminant Analizi (LDA)

LDA, boyut indirgeme ve sınıf ayrılabilirliği odaklı bir tekniktir. Veriyi, farklı sınıfları en iyi ayıran yeni boyutlarda temsil eder. LDA, her sınıfın mümkün olan en iyi şekilde ayrılması için doğrusal projeksiyonlar kullanır. Bu yöntem, denetimli öğrenme için boyutsallık azaltmaya yönelik yaygın olarak kullanılır.

* **Avantajları**

***Sınıf ayrılabilirliği:*** LDA, sınıflar arasında en belirgin ayrımı elde etmeyi hedefler ve böylece sınıflandırma doğruluğunu artırır.

***Denetimli öğrenme:*** Etiketli veri setlerinde sınıf bazlı boyutsallık azaltmak için oldukça uygundur.

* **Dezavantajları**

***Sınıflar arasında eşit kovaryans varsayımı*:** LDA, sınıfların aynı kovaryans matrisine sahip olduğunu varsayar, bu durumda bu varsayımın geçerli olmadığı durumlar hatalı sonuçlar verebilir.

***Doğrusal ilişki gerekliliği:*** *LDA* yalnızca doğrusal sınıflandırma yapar; doğrusal olmayan sınırlar için uygun değildir.

***Sınıflar arasındaki örneklem sayısı farkları:*** Sınıflar arasında büyük örneklem sayısı farkları olduğunda performans düşebilir.

* **Kütüphaneler**

***Python:***

**scikit-learn:** LinearDiscriminantAnalysis sınıfı

**numpy ve pandas:** Veri ön işleme ve yönetim

**matplotlib ve seaborn:** Sonuçların görselleştirilmesi

***R:***

**MASS paketi:** lda() fonksiyonu

**caret paketi:** LDA ile model eğitme ve doğrulama

* **Modelleme**

***Klasik LDA:*** LDA'nın standart hali, iki veya daha fazla sınıf arasındaki ayrımı maksimize eder.

***Quadratic Discriminant Analysis (QDA):*** LDA'nın daha esnek bir versiyonudur, kovaryans matrislerinin sınıflar arasında farklı olmasına izin verir.

***Regularized LDA:*** LDA'nın aşırı uyum yapmasını engellemek için düzenleme (regularization) kullanır.

* **Uygulama Alanları**

***Örüntü Tanıma:***

Yüz tanıma, el yazısı tanıma ve biyometrik analizler gibi alanlarda kullanılır.

***Yüz Tanıma:***

İnsan yüzlerini sınıflandırmak için LDA'nın kullanımı yaygındır.

***Biyoinformatik:***

Gen ekspresyonu verilerini sınıflandırmak ve analiz etmek için LDA kullanılır.

### **Bağımsız Bileşen Analizi (ICA)**

Bağımsız Bileşen Analizi (ICA), çok değişkenli sinyalleri, birbirinden bağımsız olan ve katkısal olarak birleşen bileşenlere ayırmak için kullanılan bir boyut indirgeme yöntemidir. ICA, özellikle verilerin bağımsız bileşenlerine ayrılması gerektiğinde kullanılır ve özellikle "kör kaynak ayrımı" gibi problemleri çözmek için etkilidir. Bu yöntem, her bir bileşenin, diğerlerinden istatistiksel olarak bağımsız olduğu varsayımına dayanır. Bu sayede, karmaşık veri kümelerindeki gizli bileşenleri keşfetmek mümkündür.

* **Avantajları**

***Bağımsız Bileşenler:*** ICA, veriyi bağımsız ve birbirinden ayrılmış bileşenlere ayırır, bu da veri analizini daha anlaşılır hale getirir.

***Kör Kaynak Ayrımı:*** ICA, birden fazla kaynak sinyalinin karışımından orijinal bağımsız sinyalleri çıkarabilen güçlü bir yöntemdir.

***Sinyal İşleme:*** Karmaşık ve gürültülü verilerde bile anlamlı bileşenleri ayırt edebilir.

* **Dezavantajları**

***Gürültüye Hassasiyet:*** ICA, gürültülü verilere karşı hassastır ve yanlış bileşen ayırma yapabilir.

***Doğrusal Bağımsızlık Varsayımı:*** Bileşenler arasındaki doğrusal bağımsızlık varsayımı her zaman geçerli olmayabilir.

***Veri Bağımlılığı:*** ICA, verinin doğru biçimde sunulmasını gerektirir, aksi takdirde performansı düşebilir.

* **Kütüphaneler**

***Python*:**

**scikit-learn:** FastICA sınıfı

**MNE:** Beyin görüntüleme ve sinyal işleme için ICA

**numpy ve pandas:** Veri ön işleme ve analiz

**matplotlib ve seaborn:** Sonuçların görselleştirilmesi

**R:**

**ica:** ICA uygulamaları için kullanılan popüler bir R paketidir.

**fastICA fonksiyonu:** R'deki ICA işlemlerini hızlandırır.

* **Modelleme**

***FastICA:*** ICA'nın hızlı versiyonudur, büyük veri setlerinde daha verimli çalışır.

***ICA ve Sinyal İşleme:*** Özellikle ses ve görüntü verilerinin bağımsız bileşenlere ayrılması için kullanılır.

***Özelleştirilmiş ICA Modelleri:*** ICA'nın farklı versiyonları, uygulamanın gereksinimlerine göre özelleştirilebilir, örneğin, zaman serisi verileri için adaptif ICA modelleri.

* **Uygulama Alanları**

***Sinyal İşleme:***

Ses ve görüntü verilerindeki karışık sinyalleri ayırt etmek, özellikle ses kayıplarını gidermek için kullanılır.

***Beyin Görüntüleme (EEG, fMRI):***

Beyin aktivitelerinden kaynaklanan sinyalleri bağımsız bileşenlere ayırarak nörolojik hastalıkların teşhisinde kullanılır.

**Finansal Veri Analizi:**

Piyasa verilerini ve finansal analizlerde gizli faktörleri keşfetmek için ICA kullanılır.

### UMAP (Uniform Manifold Approximation and Projection)

UMAP, çok yönlü öğrenme yöntemlerine dayalı olarak verilerin boyutlarını azaltan ve verinin yapısını koruyan bir tekniktir. Özellikle verilerin küresel ve yerel yapılarının korunmasına odaklanır, bu sayede yüksek boyutlu verilerdeki önemli yapıları daha düşük boyutlarda görselleştirmeyi sağlar. UMAP, t-SNE gibi diğer boyut indirgeme yöntemlerine kıyasla daha hızlı ve verimlidir.

* **Avantajları**

***Yerel ve Küresel Yapıyı Korur:*** UMAP, verinin hem yerel (yakın komşuluk) hem de küresel (bütün yapıyı) yapısını koruyarak yüksek kaliteli görselleştirmeler üretir.

***Verimli Hesaplama:*** Diğer boyut indirgeme tekniklerine kıyasla daha hızlıdır ve büyük veri kümelerinde daha verimli çalışır.

***Esneklik:*** Parametreler üzerinde esneklik sağlar, böylece belirli veri kümeleri için özelleştirilmiş sonuçlar elde edilebilir.

* **Dezavantajları**

***Parametre Ayarlama:***UMAP, doğru sonuçlar elde edebilmek için parametrelerin dikkatlice ayarlanmasını gerektirir.

***Yüksek Boyutlu Verilerle Uyum:*** Çok karmaşık verilerde yerel yapının kaybolması riski vardır.

***Deterministik Olmama:*** Sonuçlar, kullanılan başlatma değerlerine ve parametrelere bağlı olarak değişebilir.

* **Kütüphaneler**

***Python:***

**umap-learn:** UMAP'ı Python ile uygulamak için kullanılan popüler bir kütüphanedir.

**scikit-learn:** Boyut indirgeme için yardımcı fonksiyonlar sunar ve UMAP ile kombin edilebilir.

***R:***

**ggplot2:** UMAP çıktılarının görselleştirilmesinde kullanılabilir.

* **Modelleme**

***UMAP ve Veri Görselleştirme:*** UMAP, yüksek boyutlu verileri 2D veya 3D uzaya indirgemek için güçlü bir görselleştirme aracıdır.

***UMAP ve Kümeleme:*** Kümeleme algoritmaları ile birleştirildiğinde, veri kümesindeki alt grupları keşfetmek ve görselleştirmek mümkündür.

***UMAP ve Desen Tanıma:*** UMAP, desen tanıma uygulamalarında kullanılan verileri daha anlaşılır hale getirebilir.

* **Uygulama Alanları**

***Veri Görselleştirme:***

Yüksek boyutlu verileri görselleştirerek önemli desenlerin ve yapısal bilgilerin keşfedilmesini sağlar.

***Kümeleme:***

UMAP, veri kümesindeki alt kümeleri görselleştirerek kümeleme algoritmalarına yardımcı olabilir.

***Desen Tanıma:***

Veri kümesindeki benzerlikleri ve ilişkileri ortaya çıkararak desen tanıma işlemlerine katkı sağlar.

UMAP, çok yönlü öğrenme tekniklerini kullanarak verinin önemli yapısını koruyarak, özellikle görselleştirme, kümeleme ve desen tanıma gibi alanlarda yüksek performans sunar. Hem yerel hem de küresel yapıyı koruma yeteneği ile dikkat çeker.

## ANOMALİ TESPİTİ

Anomali tespiti, denetimsiz öğrenme yöntemleriyle, çoğunluktan büyük ölçüde farklı olan veri noktalarını bulmayı amaçlayan bir tekniktir. Bu, hatalar, sahtekarlıklar veya olağandışı durumların tespiti için kullanılır. Anomali tespiti, genellikle veri kümesinde normalden sapmaları belirleyerek, potansiyel sorunları erken aşamada fark etmeye yardımcı olur.

### **Yerel Aykırı Değer Faktörü (LOF - Local Outlier Factor)**

LOF, bir veri noktasının yerel yoğunluğunu çevresindeki diğer veri noktalarıyla karşılaştırarak anomalileri tespit eder. Bu yöntem, verilerin yerel yapısını dikkate alır, böylece normalden farklı ancak küme içinde doğru olan noktaları yanlışlıkla aykırı olarak işaretlemez.

* **Avantajları**

***Yerel Duyarlılık:*** LOF, verilerin yerel yapısına duyarlıdır ve küme içindeki lokal anormallikleri daha doğru bir şekilde tespit eder.

***Esneklik:*** Farklı veri kümeleri için etkili çalışabilir.

***Çok Yönlülük:***Sadece belirli veri noktalarını değil, aynı zamanda bir kümedeki yerel yoğunluk farklarını da analiz eder.

* **Dezavantajları**

***Hesaplama Zorluğu:*** Büyük veri kümelerinde LOF hesaplaması oldukça pahalı olabilir, çünkü her bir nokta için çevresindeki yoğunluklar hesaplanmalıdır.

***Parametre Ayarı:*** LOF algoritmasının doğru sonuçlar vermesi için uygun parametrelerin seçilmesi gerekmektedir.

* **Kütüphaneler**

***Python:***

**scikit-learn:** LOF'yi uygulamak için yaygın kullanılan bir Python kütüphanesidir.

***R:***

**DMwR:** LOF algoritmasının R'deki uygulaması için kullanılabilir.

* **Modelleme**

***Yerel Yoğunluk Karşılaştırması:*** LOF, bir veri noktasının çevresindeki diğer veri noktalarının yoğunluğuyla karşılaştırarak, önemli ölçüde daha düşük yoğunluklu noktaları aykırı olarak işaretler.

***Küme Dışı Tespit:*** Bu yöntem, özellikle sıkı kümeler içinde anormallikleri tespit etmek için uygundur.

* **Uygulama Alanları**

***Sahtekarlık Tespiti:***

Finansal işlemlerde olağandışı aktivitelerin ve dolandırıcılığın tespiti için kullanılır.

***Sağlık İzleme:***

Medikal verilerde olağandışı semptomlar veya hatalı ölçümler tespit edilerek, potansiyel sağlık sorunları önceden tespit edilebilir.

***İnternet Güvenliği:***

Ağ trafiğinde olağandışı aktiviteleri tespit etmek için kullanılır.

### **İzolasyon Ormanı (Isolation Forest)**

İzolasyon Ormanı, anomalileri tespit etmek için kullanılan farklı bir yaklaşımdır. Bu algoritma, veri noktalarını özelliklerine göre özyinelemeli olarak izole etmeye çalışır. Anomaliler, ortalama bir normal veri noktasına göre daha az adım gerektirdiği için kolayca izole edilebilir. Bu algoritma, veri noktalarının hızlı bir şekilde ayrılmasını sağlar.

* **Avantajları**

***Verimli Performans:*** İzolasyon Ormanı, büyük veri kümelerinde çok hızlı çalışabilir ve düşük hesaplama kaynaklarıyla etkili sonuçlar verebilir.

***Düşük Bellek Tüketimi:*** Algoritma, belleği verimli kullanarak yüksek verimlilik sağlar.

***Doğrudan Aykırı Değer Tespiti:*** Veriler doğrudan ve hızlı bir şekilde izole edilerek aykırı değerler tespit edilir.

* **Dezavantajları**

***Aykırı Değerlerin Yanlış Sınıflandırılması:*** İzolasyon Ormanı, çok yoğun veri kümelerinde ya da düşük etkileşimli veri kümelerinde, normal noktaları yanlışlıkla anormal olarak sınıflandırabilir.

***Parametre Ayarı:*** Doğru parametrelerin seçilmesi gerekebilir, özellikle veri noktalarının yoğunluğuna göre.

* **Kütüphaneler**

***Python:***

**scikit-learn:** İzolasyon Ormanı algoritmasını uygulamak için yaygın kullanılan bir Python kütüphanesidir.

***R:***

**isofor:** İzolasyon Ormanı algoritmasını R'de uygulamak için kullanılabilir.

* **Modelleme**

***Özyinelemeli İzolasyon:*** İzolasyon Ormanı, veri noktalarını özyinelemeli olarak izole eder ve anormal noktaları hızlıca tanımlar. Anomaliler genellikle daha az adımda izole edilebileceği için, bu noktalar daha kolay ayrılır.

***Karar Ağaçları Kullanımı:*** İzolasyon Ormanı, karar ağaçları kullanarak her bir veri noktasını izole eder ve anormallikleri tespit eder.

* **Uygulama Alanları**

***Finansal Anomali Tespiti:***

Bankacılık işlemlerinde dolandırıcılık veya olağandışı hareketler için kullanılır.

***Çevre İzleme:***

Çevre verilerinde anormal değişiklikleri tespit etmek için kullanılabilir.

***Sağlık Takibi:***

Sağlık verilerinde, anormal hastalık belirtilerinin tespiti için uygulanabilir.

## **APRİORİ ALGORİTMASI**

Apriori algoritması, işlem verilerinde sık görülen öğe kümelerini belirlemek ve bu kümeler arasında ilişki kuralları üretmek için kullanılan bir algoritmadır. Bu algoritma, veritabanlarında yer alan öğeler arasındaki bağlantıları ve ilişkileri keşfederek, alışveriş sepeti analizlerinde yaygın olarak kullanılır.

* **Avantajları**

***Basit ve Uygulaması Kolay:***Algoritmanın temel prensibi oldukça basittir ve uygulanması kolaydır.

***Büyük Veri Kümeleri İşlenebilir:*** Apriori algoritması, büyük veri kümelerinde de kullanılabilir.

***İlişkiler ve Kurallar:*** Sık kullanılan öğe kümeleri arasında güçlü ilişki kuralları oluşturulabilir.

* **Dezavantajları**

***Büyük Öğeler İçin Hesaplama Zorluğu:*** Özellikle büyük öğe kümeleri söz konusu olduğunda, algoritma hesaplama açısından pahalıdır.

***Zaman ve Bellek Tüketimi:*** Veritabanındaki tüm kombinasyonları gözden geçirdiği için zaman ve bellek tüketimi yüksek olabilir.

* **Kütüphaneler**

***Python:***

**mlxtend:** Apriori algoritmasının Python implementasyonu için popüler bir kütüphanedir.

***R:***

**arules:** R için Apriori algoritması ve ilişki kuralları çıkarma aracı.

* **Modelleme**

***Alışveriş Sepeti Analizi:*** Apriori, ürünlerin hangi sıklıkta birlikte alındığını analiz eder, bu da çapraz satış stratejileri için faydalıdır.

***İlişki Kuralları Çıkartma:*** Apriori, öğe kümeleri arasındaki ilişkileri analiz ederek, bu ilişkiler üzerinden pazarlama stratejileri geliştirilmesine olanak sağlar.

* **Uygulama Alanları**

***Pazar Sepeti Analizi:***

Müşterilerin birlikte satın aldığı ürünleri belirleyerek, çapraz satış stratejileri geliştirir.

***Çapraz Satış Stratejileri:***

Çeşitli ürünlerin birlikte satılmasını öneren stratejiler oluşturur.

***Web Kullanım Madenciliği:***

Kullanıcıların web sitelerinde gezindikleri sayfalar arasındaki ilişkileri analiz eder.

## **TORBALAMA (BAGGING)**

Bagging, özellikle karar ağaçları kullanılarak, verilerin farklı alt kümeleri üzerinde eğitilen birden fazla modelin tahminlerinin ortalamasını alarak varyansı azaltmaya yardımcı olan bir topluluk öğrenme yöntemidir. Bu yöntem, modelin aşırı uyum yapmasını engellemeye yönelik tasarlanmıştır.

1. **Avantajları**

***Aşırı Uyumu Azaltır*:** Farklı veri alt kümelerinde yapılan eğitim, modelin aşırı uyum yapmasını engeller.

***Stabiliteyi Artırır:*** Bagging, modelin tahmin performansını daha istikrarlı hale getirir.

***Yüksek Doğruluk*:** Birden fazla modelin tahminlerinin birleştirilmesi, doğruluğu artırabilir.

* **Dezavantajları**

***Daha Fazla Kaynak Gereksinimi:*** Birden fazla model eğitildiği için, daha fazla hesaplama kaynağına ihtiyaç duyulabilir.

***Model Karmaşıklığı:*** Birden fazla modelin birleşimi, modelin açıklanabilirliğini zorlaştırabilir.

* **Kütüphaneler**

***Python:***

**scikit-learn:** Bagging için karar ağaçları ve diğer modellerin kullanılması mümkündür.

***R:***

**randomForest:** Bagging tabanlı rastgele orman algoritması için yaygın bir kütüphanedir.

* **Modelleme**

***Aşırı Uyumu Azaltma:*** Bagging, özellikle karar ağaçlarıyla birlikte kullanıldığında, veri setindeki rastgeleliklerin etkisini azaltarak modelin doğruluğunu artırır.

***Topluluk Yöntemi:*** Birden fazla modelin kararlarının birleştirilmesi, genellikle tek bir modelin kararından daha iyi sonuçlar verir.

* **Uygulama Alanları**

***Finansal Tahmin:***

Yatırım ve kredi değerlendirmesi gibi finansal kararlar için tahminler yapılır.

***Görüntü Tanıma:***

Görüntü verilerinde çeşitli modellerin birleştirilmesi ile doğruluk artırılır.

***Sağlık Teşhisi:***

Hastalık teşhisi ve tahmin süreçlerinde Bagging kullanılarak doğruluk ve güvenilirlik artırılır.

## **ECLAT ALGORİTMASI**

Eclat algoritması, sık karşılaşılan öğe kümelerini keşfetmek için derinlemesine arama yöntemini kullanarak, veritabanı taramalarının sayısını azaltır ve böylece verimliliği artırır. Dikey veri formatını kullanarak, Apriori algoritmasından daha verimli çalışır.

* **Avantajları**

***Verimli:*** Veritabanı taramaları azaltıldığından, Apriori'ye kıyasla daha verimlidir.

***Dikey Veri Formatı Kullanımı:*** Veritabanı üzerinde daha hızlı işlem yapılmasını sağlar.

***Bellek Kullanımı*:** Dikey veri formatı sayesinde bellek kullanımı daha etkin hale gelir.

* **Dezavantajları**

***Bellek Tüketimi:*** Büyük öğe kümeleriyle çalışırken yeterli belleğe ihtiyaç duyulabilir.

***Karmaşık Veri Setlerinde Zorluklar*:** Özellikle karmaşık ve büyük veri setlerinde verimli sonuçlar almak zor olabilir.

* **Kütüphaneler**

***Python:***

**mlxtend:** Eclat algoritması da bu kütüphanede uygulanabilir.

***R:***

**arules:** R için Eclat algoritması implementasyonu.

* **Modelleme**

***Veritabanı Tarama:*** Eclat, veritabanındaki öğe kümeleri arasındaki ilişkileri hızlıca belirler ve işleme süresini önemli ölçüde azaltır.

***İlişki Kuralları Çıkartma:*** Öğelerin sıklığı ve ilişkilerini tespit ederek, pazarlama stratejilerini destekler.

* **Uygulama Alanları**

***Pazar Sepeti Analizi:***

Hızlı ve verimli bir şekilde alışveriş sepeti analizi yaparak, çapraz satış fırsatları belirler.

***Biyoenformatik:***

Genetik veri kümeleri üzerinde analiz yaparak biyolojik ilişkileri keşfeder.

***Metin Madenciliği:***

Metin verilerindeki sık kullanılan kelimeler ve kavramlar arasındaki ilişkileri keşfeder.

## Derin öğrenme nedir ?

Derin öğrenme, bilgisayarlara insanlar için doğal olan şeyi yapmayı öğreten bir makine öğrenme tekniğidir: “örneklerle öğrenmek”. Derin öğrenme**,** sürücüsüz araçların arkasındaki kilit teknolojidir ve bir dur işaretini tanımalarını ya da bir yayayı elektrik direğinden ayırt etmelerini sağlar. Telefonlar, tabletler, TV’ler ve kablosuz hoparlörler gibi tüketici cihazlarında ses kontrolünün anahtarıdır.

## Derin öğrenme nasıl çalışır?

Derin öğrenme, insan beyninin çalışma şekline göre genel hatlarıyla modellenen algoritmalar olan [sinir ağlarının](https://blogs.oracle.com/bigdata/post/neural-networks-in-deep-learning) katmanları tarafından desteklenir. Büyük miktarlarda veri ile eğitim, sinir ağındaki nöronları konfigüre etmektir. Sonuç, eğitildikten sonra yeni verileri işleyen derin öğrenme modelidir. Derin öğrenme modelleri, birden fazla veri kaynağından bilgi alır ve bu verileri insan müdahalesine gerek kalmadan gerçek zamanlı olarak analiz eder. Derin öğrenmede, grafik işleme birimleri (GPU'lar), aynı anda birden fazla hesaplamayı işleyebildikleri için eğitim modellerine yönelik olarak optimize edilmiştir.

Derin öğrenme, otomasyonu ve analitik görevleri iyileştirebilen birçok [yapay zekâ (AI) teknolojisine](https://www.oracle.com/tr/artificial-intelligence/) yön veren kavramdır. Çoğu kişi internete göz attığında veya cep telefonlarını kullandıklarında her gün derin öğrenmeyle karşılaşırlar. Diğer sayısız uygulama yazılımının yanı sıra, derin öğrenme, YouTube videoları için alt yazı oluşturmak, telefonlarda ve akıllı hoparlörlerde konuşma tanıma gerçekleştirmek, fotoğraflar için yüz tanıma sağlamak ve [kendi kendine giden arabaları](https://blogs.oracle.com/oracle-france/deep-learning-definition) sunmak için kullanılır.

## Yapay Sinir Ağı Nedir?

Yapay sinir ağları (YSA), insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri, herhangi bir yardım almadan otomatik olarak gerçekleştirebilmek amacı ile geliştirilen bilgisayar sistemleridir.

Yapay sinir ağları insan beyni örnek alınarak, öğrenme sürecinin matematiksel olarak modellenmesi sonucu ortaya çıkmıştır. Beyindeki biyolojik sinir ağlarının yapısını, öğrenme, hatırlama ve genelleme kabiliyetlerini taklit eder. Yapay sinir ağlarında öğrenme işlemi örnekler kullanılarak gerçekleştirilir. Öğrenme esnasında giriş çıkış bilgileri verilerek, kurallar koyulur.

### Yapay Sinir Ağının Yapısı

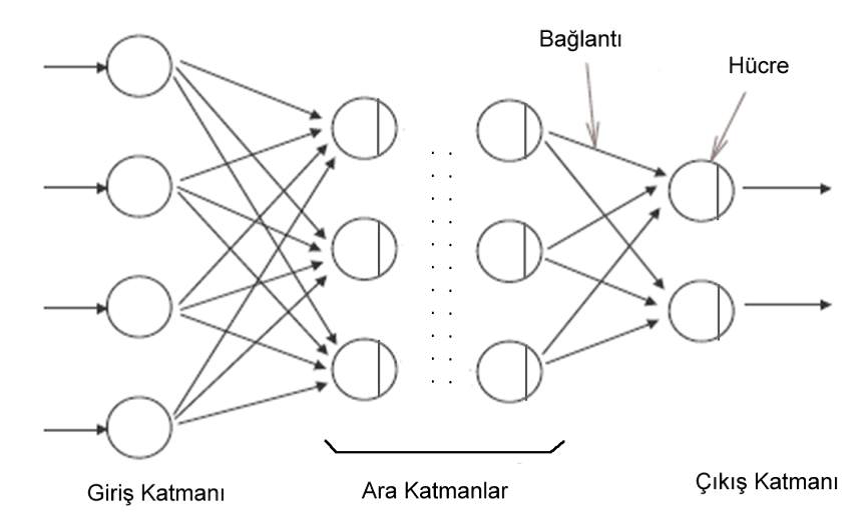
Yapay sinir ağları yapay sinir hücrelerinin birbirine bağlanmasıyla oluşan yapılardır. Yapay sinir ağları tıpkı insan beyni gibi nöronlardan oluşur. Tüm nöronlar birbirine bağlıdır ve çıktıyı etkilemektedir. Yapay sinir ağları üç ana katmanda incelenir;

* Giriş Katmanı,
* Ara (Gizli) Katmanlar
* Çıkış Katmanı.

İlk katman olan input layer’ da gözlemler sisteme aktarılır. Bu katmandaki düğüm sayısı, gözlemi en iyi şekilde temsil edecek olan öznitelik sayısına eşittir. Giriş katmanında değerler kopyalanarak birden fazla katmana gönderilebilir. Aktarım aşamasında değerlere herhangi bir işlem uygulanmaz. Bizim örneğimizde, giriş katmanında dört nöron var: Kalkış Havaalanı, Varış Havaalanı, Kalkış Tarihi ve Firma. Giriş katmanı, girişleri ilk gizli katmana gönderir.

Gizli katmanlar girdilerimizde matematiksel hesaplamalar yapar. Yapay sinir ağları oluşturmadaki zorluklardan biri, her bir katman için nöronların sayısının yanı sıra gizli katmanların sayısına da karar vermektir. Bu düğümlerde belirli threshold koşulları uygulanarak output sayısı kadar değer elde edilebilir.

output layer(çıkış katmanı çıktı verilerini döndürür. Bizim örneğimizde, bize fiyat tahmini verir.



**Şekil 19: Yapay Sinir Ağları Katmanları**

Her nöron bir Aktivasyon Fonksiyonuna sahiptir. Aktivasyon fonksiyonunun amaçlarından biri nörondan elde edilen çıktıları “standartlaştırmak” tır.

Bir veri kümesi sinir ağının tüm katmanlarından geçtikten sonra, çıktı katmanından sonuç olarak döner.

## Evrişimsel Sinir Ağları (Convolutional Neural Network-CNN)

Evrişimsel Sinir Ağları (CNN), derin öğrenme alanında yaygın olarak kullanılan ve özellikle görüntü işleme, nesne tanıma, yüz tanıma, doğal dil işleme gibi uygulamalarda kullanılan bir derin öğrenme modelidir. CNN'ler, biyolojik görme sisteminden ilham alınarak geliştirilmiştir ve verinin yerel özelliklerini öğrenmeye odaklanır. Google ve Facebook gibi şirketler başta olmak üzere birçok büyük şirket tarafından sıkça kullanılmaktadır.

### CNN Katmanları:

* ***Evrişim (Convolutional) Katmanları:***
* Giriş verisi üzerinden belirli filtrelerle (kernels) kaydırılarak özelliklerin çıkarılması sağlanır.
* Görüntünün belirli bölgelerindeki kenarlar, köşeler, dokular gibi temel özellikleri çıkarır.
* ***Havuzlama Katmanları (Pooling Layers):***

1. Boyut indirgeme işlemi yapar ve modelin hesaplama yükünü azaltır.
2. Max pooling veya average pooling gibi yöntemlerle özellik haritalarını küçültür.

* ***Düzleştirme Katmanı (Flattening Layers):***
* Evrişimli ve pooling katmanlarından elde edilen özellik haritaları (feature maps) çok boyutludur. Flatten katmanı bu çok boyutlu özellik haritalarını tek boyutlu bir vektöre dönüştürür. Bu, özellik haritalarını tam bağlantılı katmanlara uygun hale getirir.
* ***Tam Bağlantılı (Fully Connected) Katmanlar:***
* Görüntüden çıkarılan özellikleri sınıflandırma veya regresyon gibi bir çıktıya dönüştürür.

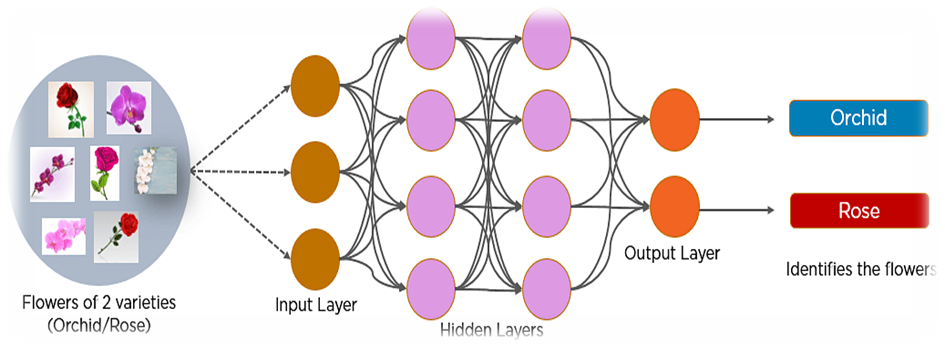
### CNN'nin Avantajları:

* **Özellik Öğrenimi:** Görüntü veya veri üzerindeki özellikleri otomatik olarak öğrenir, manuel özellik mühendisliğine gerek kalmaz.
* **Uzaysal Hiyerarşi:** Girdinin yerel ve küresel özelliklerini hiyerarşik bir şekilde öğrenir.
* **Verimlilik:** Diğer yapay sinir ağı türlerine kıyasla, özellikle görüntü verileri üzerinde daha hızlı ve verimli çalışır.

### Kullanım Alanları:

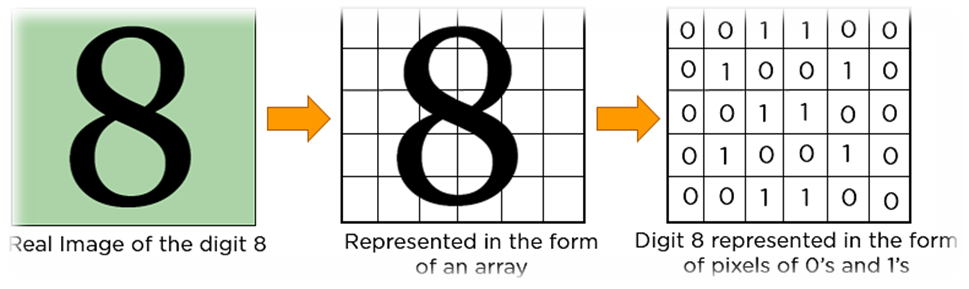
* Görüntü sınıflandırma (ör. yüz tanıma, nesne algılama).
* Tıbbi görüntü analizi (ör. MR taramalarının analizi).
* Video işleme ve eylem tanıma.
* Doğal dil işleme (ör. kelime gömüleri ve dil modelleme).
* Ses ve zaman serisi analizi.

## CNN Nasıl Çalışır ?



**Şekil 20: CNN Örneği**

Görselde, iki tür çiçek (Orkide ve Gül) örnekleri gösterilmektedir. Bu çiçek görüntüleri yapay sinir ağına giriş olarak verilir. Bu görüntüler, model tarafından sayısal verilere (piksel değerlerine) dönüştürülerek işlenir. Giriş verileri, birden fazla gizli katmandan geçer. Gizli katmanlar, modelin çiçeklerin özelliklerini (renk, şekil, desen gibi) öğrenmesini sağlar. Bu katmanlar, giriş verileri üzerinde matematiksel işlemler yaparak özellikleri daha soyut seviyelere çıkarır. Katmanlar arasındaki bağlantılar, ağırlıklar (weights) ve aktivasyon fonksiyonları aracılığıyla bilgiyi işler. Model, girişte verilen görüntüye göre bir tahminde bulunur. Çıkış katmanında, Orchid (Orkide) ve Rose (Gül) olmak üzere iki sınıf vardır. Görsel, modelin bir görüntüyü analiz edip hangi çiçek türüne ait olduğunu belirlemesini göstermektedir.



**Şekil 21: CNN Örneği**

İlk olarak, sol tarafta '8' rakamının gerçek bir görüntüsünü görüyoruz. Bu, herhangi bir dijital ortamdan alınmış olabilir, örneğin bir el yazısı veya bir yazı tipiyle oluşturulmuş bir görsel. Bu aşama, bilgisayarın henüz anlamlandıramadığı ham bir veriyi temsil eder.

İkinci aşamada, bu görsel bir ızgaraya (matriks) dönüştürülüyor. Her bir hücre, görüntünün küçük bir parçasını temsil ediyor. Bu adımda görsel, bilgisayarın daha kolay işlem yapabileceği bir formata getiriliyor.

Son aşamada, '8' rakamı, piksellerden oluşan bir matris olarak ifade ediliyor. Burada:

* '1' değerleri siyah pikselleri, yani rakamın kendisini temsil eder.
* '0' değerleri ise beyaz pikselleri, yani arka planı temsil eder.

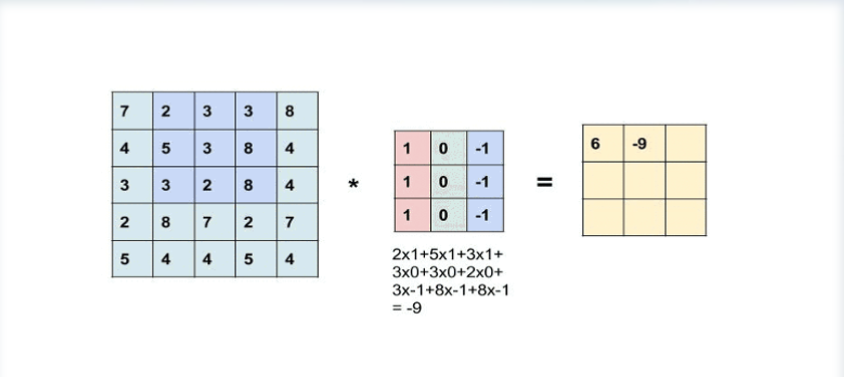
Bu, görselin bilgisayar tarafından tamamen sayısal bir formatta anlaşılabilir hale geldiği aşamadır.

### Peki bu süreç neden önemli?

Çünkü bilgisayarlar görsel bilgiyi doğrudan anlayamaz. Bu dönüşüm, rakamların veya görsellerin bilgisayarın analiz edebileceği ve öğrenebileceği bir formata dönüştürülmesini sağlar. Örneğin, karakter tanıma, el yazısı tanıma veya görüntü sınıflandırma gibi işlemlerde bu tür bir dönüşüm temel bir adımdır.

* ***Evrişim Katmanı***

Bu katmanda girdi olarak gelen resim bir filtreden geçirilir. Filtreden geçirilme sonucu oluşan değerler öznitelik haritasını oluşturur.



**Şekil 22: CNN Örneği**

Sol tarafta, bir görüntünün sayısal temsili olan bir matris görüyoruz. Her hücre, görüntünün bir piksel değerini temsil eder. Bu örnekte, 5x5 boyutunda bir giriş matrisimiz var ve her bir sayı, pikselin parlaklık değerini ifade ediyor.

Ortada, 1x1 veya daha büyük boyutlarda olabilen bir çekirdek (kernel) matrisi yer alıyor. Bu çekirdek, görüntüden belirli özellikleri çıkarmak için kullanılır. Örneğin, bu görselde 3x3 boyutunda bir kernel kullanılmış ve kenar tespiti gibi özellikleri öğrenmeye yarayabilecek değerlerle doldurulmuş (örneğin, 1, 0, -1).

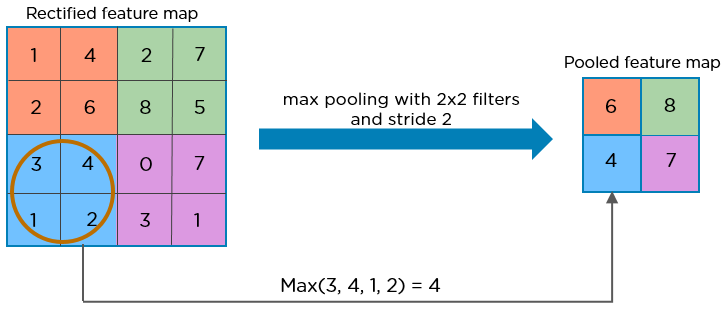
Evrişim işlemi, çekirdek matrisinin giriş görüntüsü üzerinde gezdirilmesiyle yapılır. Burada:

Çekirdek, giriş matrisinin belirli bir bölgesiyle çarpılır.

Çarpımların toplamı alınır ve bu toplam yeni bir çıktı matrisine yerleştirilir.

* ***Havuzlama Katmanı***

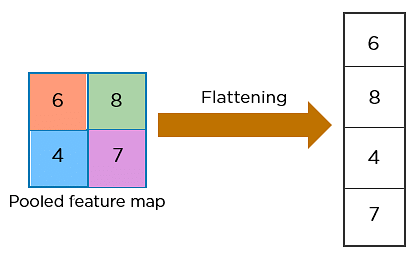
Convolutional katmanından sonraki ikinci katman havuzlama katmanıdır.



**Şekil 23**

Pooling layer genellikle oluşturulan öznitelik matrislerine uygulanır. Şekil 4'de 4x4 lük bir Öznitelik matrisi matrisi görülmektedir. Burada 2x2’lik bir maksimum pooling uygulanırsa sağdaki 2x2’lik matris oluşur. Öznitelik matrisinde pooling için belirlenen boyutlarda (Şekil 4 için 2x2) matrisler oluşturularak, o matris içerisindeki en büyük değer alınır. Turuncu olarak gösterilen ilk matriste 6, yeşil olarak belirlenen matriste ise bu değer 8’dir. Genellikle kullanılan yöntem maksimum pooling olsa da mean pooling ve minimum pooling gibi yöntemler de mevcuttur. Mean pooling yönteminde ortalama alınırken, minimum pooling yönteminde ise en düşük değer alınarak işlem gerçekleştirilir.

* ***Düzleştirme Katmanı***



**Şekil 24: Düzleştirme Katmanı Örneği**

Yukarıdaki katmanlar modelde birden fazla kez kullanılabilir. Bu işlemler tamamlandıktan sonra elde edilen matris Fully Connected katmanında kullanılabilmesi için düzleştirilmesi gerekir. Bu sebepten ötürü Şekil 5'de örnek olarak görülebilecek flattening işlemi gerçekleştirilir. Bu aşamadan sonra Fully Connected Layer’ın inputları hazırlanmış olur.

## Tekrarlayan Sinir Ağları (Recurrent Neural Network-RNN)

Tekrarlayan sinir ağı veya RNN, sıralı veya zaman serisi veriler üzerinde eğitilen ve  sıralı girdilere dayalı sıralı tahminler veya sonuçlar çıkarabilen bir [makine öğrenimi](https://www.ibm.com/topics/machine-learning) (ML) modeli oluşturan derin bir [sinir ağıdır .](https://www.ibm.com/topics/neural-networks)Bir RNN, geçmiş günlük sel, gelgit ve meteoroloji verilerine dayanarak günlük sel seviyelerini tahmin etmek için kullanılabilir. Ancak RNN'ler ayrıca dil çevirisi, [doğal dil işleme (NLP)](https://www.ibm.com/topics/natural-language-processing) , [duygu analizi](https://www.ibm.com/topics/sentiment-analysis) , [konuşma tanıma](https://www.ibm.com/topics/speech-recognition)  ve resim altyazılama gibi sıralı veya zamansal sorunları çözmek için de kullanılabilir.

### RNN'nin Temel Özellikleri

* ***Hafıza Özelliği****:*

RNN'ler, ağın geçmişte aldığı girdileri hatırlayarak bu bilgiyi mevcut çıktı tahminlerinde kullanabilir.

Bu, ağın bir tür "hafıza"ya sahip olmasını sağlar.

* ***Geri Besleme Döngüsü****:*

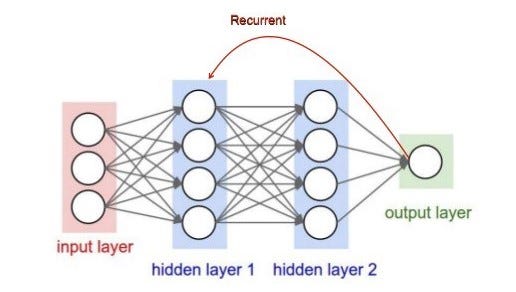
RNN'lerde döngüsel bağlantılar bulunur. Bu döngüsel yapı, önceki zaman adımlarındaki bilgiyi sonraki adımlarda yeniden kullanma imkânı sağlar.

* ***Zaman Bağımlılığı****:*

RNN'ler, girdilerin sırasının önemli olduğu görevlerde etkilidir, çünkü sıralı bilgiyi işler.

### RNN’nin Yapısı

Tekrarlayan sinir ağlarının yapısını şu örnek üzerinden anlamamız kolaylaşacaktır. İnsanlar düşüncelerini önceden öğrendiği bilgilere dayanarak oluşturur. Örneğin bir cümleyi okuduğunuzda her kelimeyi bir önceki kelimeyi baz alarak anlarız yani önceki kelimeyi hatırlayarak sonraki kelime veya kelimeleri anlamlandırırız. Geleneksel sinir ağları böyle bir şey yapamaz ve bu önemli bir eksikliktir. RNN ise bu tür sorunları ele almaktadır. Tekrarlayan sinir ağı fikri 1980’lerde Basit Tekrarlayan Ağ (Simple Recurrent Network) olarak atılmıştır. Sonraki zamanda ise Jeff Elman tarafından 1990’larda RNN önerilmiştir. Tekrarlayan sinir ağındaki temel düşünce sıralı bilgileri kullanmaktır. Geleneksel bir sinir ağında tüm girdiler ile çıktıların birbirinden bağımsız olduğunu varsayılmaktadır. Örneğin, bir cümle içinde bir sonraki kelimeyi tahmin etmek için, hangi sözcüklerin o anki kelimeden önce geldiğini bilmek gerekmektedir. RNN mimarisinin yinelenen olarak adlandırılmasının sebebi, bir dizinin her öğesi için aynı görevi önceki çıktılara bağlı olarak yerine getirmesidir. Şekil 1’de RNN yapısını görülmektedir.



**Şekil 25**

Bir giriş katmanına, iki gizli katmana ve bir çıkış katmanına sahiptir. Bu katmanların hepsi bağımsız şekilde çalışmaktadır. Dolayısıyla her katmanın kendine göre bir işlevi vardır ve her katman farklı bir işlevi yerine getirir. Daha da detaylandırılması gerekirse her katmandaki yapıların ağırlıkları vardır ve katmana özgü şekilde eşik değerleri belirlenir. Bu sayede sistemin daha olası sonuçlar vermesi amaçlanır. Tekrarlanan bu aşamalar sonucunda bir önceki giriş durumunun depolanır ve yeni elde edilen giriş değeriyle birleştirilir bu sayede yeni elde edilen girişin bir önceki girişle olan ilişkisi sağlanır.

### RNN'nin Avantajları

***Zaman Serisi ve Sıralı Veriyle Uyumlu***

RNN, geçmiş girişleri hatırlama yeteneği sayesinde zaman serisi, metin, konuşma ve diğer sıralı veri türleriyle etkili bir şekilde çalışabilir.

Veri noktaları arasındaki bağımlılıkları modelleyebilir. Örneğin, bir cümledeki kelimeler arasındaki ilişkiyi öğrenebilir.

***Dinamik Uzunlukta Veri Girişleri***

* Sabit uzunlukta giriş gerektiren bazı modellere (örneğin, geleneksel sinir ağları) kıyasla, RNN'ler değişken uzunlukta dizileri işleyebilir.
* Bu, özellikle metin veya konuşma tanıma gibi alanlarda faydalıdır.

***Paylaşımlı Parametreler***

* Zaman boyunca aynı ağırlıkların kullanılması (parametre paylaşımı), modelin daha az parametreyle öğrenmesini sağlar. Bu, hesaplama açısından verimlilik sunar.

***Sürekli Veri Üzerinde Kullanım***

* RNN'ler, sürekli veri akışlarında (örneğin, gerçek zamanlı analiz) kullanılabilir. Bu, anlık veri işleme ve tahmin gerektiren uygulamalarda avantaj sağlar.

**RNN'nin Zorlukları**

* ***Vanishing Gradient (Kaybolan Gradyan) Problemi****:*
* Uzun dizilerde gradyanlar zamanla sıfıra yaklaşabilir, bu da ağırlık güncellemelerini zorlaştırır.
* Bu sorun, LSTM ve GRU gibi gelişmiş RNN türleriyle çözülmüştür.
* ***Uzun Vadeli Bağımlılıkların Öğrenilmesi****:*
* Standart RNN'ler, uzun dizilerde önceki bilgiyi etkili bir şekilde hatırlamakta zorlanabilir.

## Tekrarlayan Sinir Ağı Uygulamaları (RNN Applications)

* ***Doğal Dil İşleme (NLP):***

**Dil Modelleme:** RNN'ler bir cümledeki bir sonraki kelimeyi tahmin edebilir; bu da metin oluşturma ve otomatik tamamlama gibi görevler için faydalıdır.

**Makine Çevirisi** : RNN'ler çeviri görevlerinde kullanılabilir; bir RNN giriş cümlesini kodlarken diğeri hedef dilde kodunu çözer.

**Duygu Analizi** : RNN'ler, bir dizideki kelimelerden bağlamsal bilgileri yakalayarak metin duygusunu analiz edebilir.

***2****.****Konuşma Tanıma ve Sentezi:***

**Konuşmadan Metne:** RNN'ler konuşulan dili yazılı metne dönüştürmek için kullanılır ve bu da onları konuşma tanıma sistemlerinin omurgası haline getirir.

**Metinden Konuşmaya** : RNN'ler, metin girişinden insan benzeri konuşma üretebilir, böylece sesli asistanları ve erişilebilirlik araçlarını iyileştirebilir.

***3****.****Zaman Serisi Analizi ve Tahmini:***

**Finansal Tahmin** : RNN'ler hisse senedi fiyatlarını, döviz kurlarını ve diğer finansal değişkenleri tahmin edebilir.

**Hava Durumu Tahmini:** RNN'ler gelecekteki hava koşullarını tahmin etmek için geçmiş hava durumu verilerini analiz edebilir.

***4****.****Müzik Üretimi:***

RNN'ler, mevcut müzik bestelerinden kalıplar öğrenerek ve yeni besteler üreterek müzik dizileri üretebilirler.

## Derin öğrenme ve sinir ağları arasındaki fark nedir?

[Derin öğrenme](https://blogs.oracle.com/ai-and-datascience/post/what-is-deep-learning), çok katmanlı sinir ağlarına verilen addır. Fotoğraflar veya ses gibi gözlemsel verileri anlamlandırmak için sinir ağları, verileri birbirine bağlı düğüm katmanlarından geçirir. Bilgi bir katmandan geçtiğinde, o katmandaki her bir düğüm, veriler üzerinde basit işlemler gerçekleştirir ve sonuçları seçerek diğer düğümlere iletir. Sonraki her katman, ağ çıktıyı oluşturana kadar bir öncekinden daha yüksek düzeyde bir özelliğe odaklanır.

Girdi katmanı ile çıktı katmanı arasındaki katman gizli katmanlardır. Temel bir sinir ağı bir veya iki gizli katmana sahip olabilirken, bir derin öğrenme ağı düzinelerce hatta yüzlerce katmana sahip olabilir. Farklı katman ve düğümlerin sayısını artırmak ağın doğruluğunu artırabilir. Ancak, daha fazla katman ayrıca bir modelin daha fazla parametre ve hesaplama kaynağı gerektireceği anlamına da gelebilir.

Derin öğrenme, ham verileri alan bir dizi girdiye sahip olan sinir ağı katmanları aracılığıyla bilgileri sınıflandırır. Örneğin, bir sinir ağı kuş görüntüleriyle eğitilirse kuşların görüntülerini tanımak için kullanılabilir. Daha fazla katman, kargayı tavuktan ayırt etmeye kıyasla kargayı kuzgundan ayırt etmek gibi daha kesin sonuçlar sağlar. Derin öğrenme algoritmalarının temelindeki derin sinir ağları, girdi ve çıktı düğümleri arasında birkaç gizli katmana sahiptir, bu da daha karmaşık veri sınıflandırmalarını gerçekleştirebilecekleri anlamına gelir. Derin öğrenme algoritması büyük veri kümeleriyle eğitilmelidir ve ne kadar çok veri alırsa o kadar doğru olur. Yeni kuş resimlerini doğru bir şekilde sınıflandırmadan önce algoritmanın binlerce kuş resmiyle beslenmesi gerekecektir.

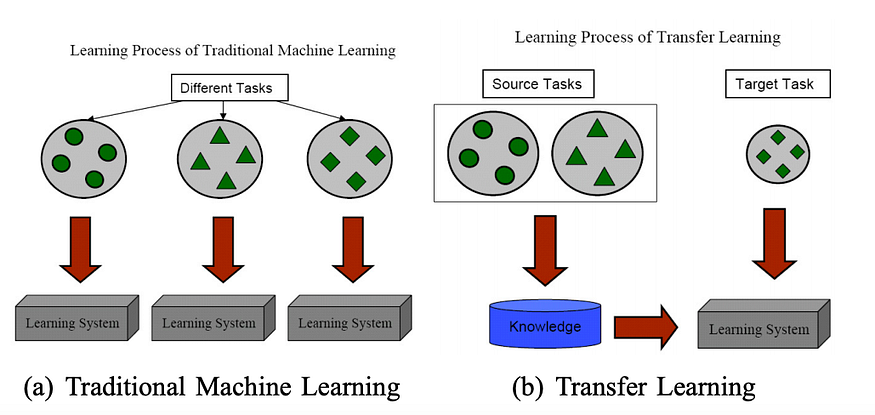
Sinir ağları söz konusu olduğunda, derin öğrenme modelinin eğitimi çok kaynak yoğunlukludur. Bu, sinir ağının, eğitim sırasında ayarlanan ağırlıklar (girdiler arasındaki bağlantının gücünü temsil eden parametreler) kullanılarak gizli katmanlarda işlenen girdileri alması ve ardından modelin bir tahmin vermesidir. Ağırlıklar, daha iyi tahminler yapmak için eğitim girdilerine göre ayarlanır. Derin öğrenme modelleri, büyük miktarda veriyi eğitmek için çok zaman harcar, bu nedenle yüksek performanslı hesaplama çok önemlidir.

## Transfer Öğrenme Nedir?

Öğrenme aktarımı,  bir problemi çözerken elde ettiği bilgiyi saklayıp, başka bir problem ile karşılaştığında o bilgiyi kullanmasıdır. Öğrenme aktarımı ile önceki bilgiler kullanılarak daha az eğitim verisi ile daha yüksek başarı gösteren ve daha hızlı öğrenen modeller elde edilir.

Transfer öğrenmesiyle, temel olarak bir görevde öğrenilenleri kullanarak başka bir görevdeki genellemeyi iyileştirmeye çalışırız. Bir ağın "görev A"da öğrendiği ağırlıkları yeni bir "görev B"ye aktarırız.

Örneğin; bisiklet kullanabilen birinin araba kullanmayı öğrenmesi bisiklet kullanamayan birine göre daha kolay ve hızlı olabilir. Birbirine benzer olan bu olaylarda kişi bisiklet sürmeyi öğrendiği sırada kazandığı araç kontrol etme becerisini araba sürerken kullanmakta ve farkında olmadan öğrenme aktarımı yapmaktadır.

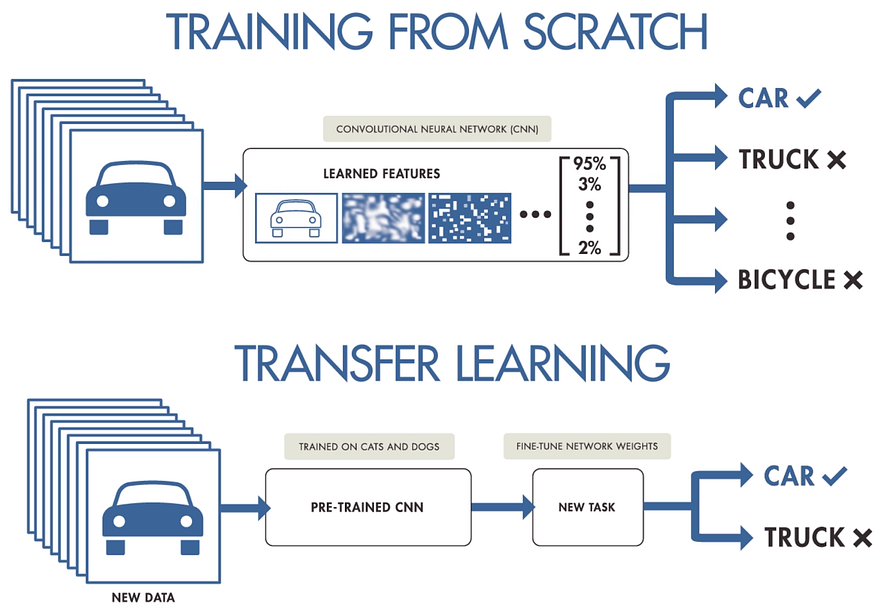


**Şekil 26: Transfer Öğrenme**

Normalde makine öğrenmesinde her bir görev için ayrı bir ‘sıfırdan öğrenme’ gerçekleştirilir. Ancak bazı görevlerden öğrenilen birtakım bilgileri başka görevlerde de kullanmak mümkün ve avantajlı olacağından kaynak görevlerden elde edilen bilgi hedef görevin çözümü için kullanılmaktadır.

Öğrenme aktarımı ile daha önceden eğitilmiş modellerden elde edilen özellikler, ağırlıklar vb. yeni bir görev için kullanılmaktadır. Bu yöntemin işe yaraması için aktarılacak bilgilerin genel bilgiler olması gerekir, yani kaynak göreve özgü olmak yerine hem kaynak hem de hedef görevler için uygun olan bilgiler aktarılır.

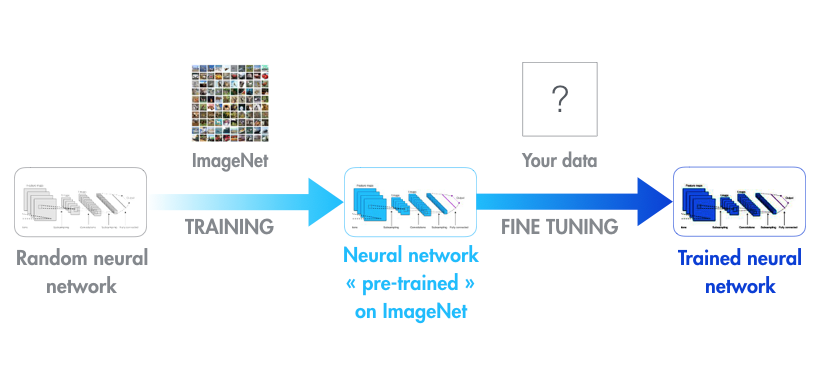
Örneğin, bilgisayarla görme alanında kedi ile köpeği ayırt edebilen bir modelde elde ettiğimiz bilgileri 2 farklı hedef görevde kullanmayı düşünelim. Birincisi motosiklet, bisiklet vb. araç tanıma için ikincisi aslan, kaplan, at vb. diğer hayvanları tanıma olsun. Birinci görev için aktarılacak bilgi tahmin edileceği üzere ikinci görev için aktarılacak bilgi ile aynı olamaz. Birinci görev için öğrenme aktarımı yapıldığında aktarılan bilgi hemen hemen her görüntü üzerinde aynı olan düşük seviyeli özellikler, kenarlar, köşeler, şekillerdir. İkinci görev için öğrenme aktarımı yapıldığında ise kedi köpek tanımada keşfedilen göz, burun, kulak gibi karmaşık şekillerin de bilgisinin aktarılması gerekir.



**Şekil 27: Transfer Öğrenme Süreci**

### Öğrenme aktarımı sürecinde 3 soru sorulur:

* ***Neyin aktarılacağı:*** Öğrenmenin aktarımı yapılırken uygulanan ilk ve en önemli aşamadır. Kaynak veriden hedefe hangi bilgi aktarılınca hedef görevin başarısının artacağına cevap aranan bölümdür. Dikkat edilmesi gereken husus ise bazı bilgilerin kullanılan veriye/göreve özgü olabileceği ve bazısının kullanılan alan ne olursa olsun ortak olabileceğidir.
* ***Ne zaman aktarılacağı:***Bu aşamada verinin hangi durumlarda aktarılacağı ya da hangi durumlarda aktarılmaması gerektiği durumları belirlenir. Bazı durumlarda kaynak veri ve hedef veri birbirinden çok farklı olabilir. Bu gibi durumlarda aktarım başarılı olmayabilir. Hatta hedef veri üzerindeki eğitimin başarısını düşürebilir. Bu duruma ‘olumsuz aktarım/negatif transfer’ denir. Q klavye kullanan birinin F klavyede yazamaması gibi.
* ***Nasıl aktarılacağı:*** Burada bilginin/öğrenmenin aktarımı için hangi yöntemlerin kullanılacağına karar verilir. Var olan yöntemlerin direkt kullanılması veya değiştirilmesi yaklaşımı izlenir.



**Şekil 28: Süreç**

Öğrenme aktarımında kaynak görevi isteğimiz bilgileri öğrenecek şekilde kendimiz geliştirebiliriz veya hazır bir model kullanabiliriz. Uygulamada daha çok tercih edilen yöntem ikincisidir. Bu tarz uygulamaların en güzel örneklerinden biri ImageNet yarışmasıdır. Bu yarışmada 1000 sınıflı fotoğrafları sınıflandırma gibi büyük ve zorlu bir görüntü sınıflandırma görevi vardır.

### Transfer Öğrenmenin Aşamaları

* ***Kaynak Model Eğitimi****:*

Büyük ve genelleştirilmiş bir veri kümesi üzerinde bir model eğitilir.

Örneğin, ResNet, VGG veya BERT gibi modeller, yaygın olarak kullanılan önceden eğitilmiş modellerdir.

* ***Hedef Göreve Aktarma****:*

Kaynak modelden, ağırlıklar ve özellik çıkarıcılar alınır.

İki farklı strateji uygulanabilir:

**Dondurma (Freezing)**: Önceden eğitilmiş katmanlar dondurularak yalnızca son katmanlar yeniden eğitilir.

**İnce Ayar (Fine-Tuning)**: Tüm ağırlıklar güncellenir ve model hedef veri kümesi için yeniden eğitilir.

### Transfer Öğrenmenin Avantajları

* ***Hızlı Model Eğitimi****:*
* Sıfırdan bir model eğitmekten daha az zaman ve hesaplama gücü gerektirir.
* ***Sınırlı Veri Kullanımı****:*
* Hedef görev için yeterli veri bulunmasa bile başarılı sonuçlar elde edilebilir.
* ***Genelleştirilebilirlik****:*

Kaynak görevde öğrenilen özellikler, genelde başka görevlerde de faydalı olur (örneğin, görüntü verisinde kenar algılama gibi temel özellikler).

### Transfer Öğrenme Yöntemleri

Öğrenme aktarımı kaynak/hedef öğrenme görev ve verilerine göre üç grupta ifade edilir.

#### Tümevarımsal Öğrenme Aktarımı (Inductive Transfer Learning)

Hedef görevin kaynak görevden farklı olduğu durumlardır. Görevlerdeki verilerin etiketli veya etiketsiz olması bakımından ikiye ayrılır.

Birincisi kaynak görevde fazla miktarda etiketli verinin bulunmasıdır. Bu durumda, öğrenme aktarımı çok yönlü öğrenme (multi-task learning) gibi çalışır. Ancak çok yönlü öğrenmenin eğitildiği tüm görevlerde başarı göstermesi beklenirken öğrenme aktarımında amaç hedef görev başarısının yüksek olmasıdır. Kaynak görev için geliştirilen bir model hedef görev için başlangıç noktası olarak kullanılır. Örneğin, kedi köpek sınıflandırma için kullanılan modelin bilgilerini araç sınıflandırma kullanmak.

İkincisi ise kaynak verinin etiketli veri içermemesi durumudur. Bu durumda tümevarımsal öğrenme aktarımı bu sefer kendi kendine öğrenme (self taught learning) gibi çalışır. Kendi kendine öğrenmede kaynak ve hedef verilerin etiket uzayı farklıdır, bu da kaynak verisinin doğrudan kullanılamayacağı anlamına gelir. Örneğin, ne olduğu bilinmeyen resimlerden çıkartılan bilgilerin araç sınıflandırma için kullanılmasıdır.

#### Dönüştürücü Öğrenme Aktarımı (Transductive Transfer Learning)

Kaynak ve hedef görevlerin aynı fakat verilerinin farklı olmasıdır. Hedef görevde etiketli verinin bulunmadığı fakat kaynak görevde bol miktarda etiketli verinin olduğu durumlarda kullanılır. Hedef ve kaynak veri kümelerine göre dönüştürücü öğrenme aktarımı iki farklı şekilde ifade edilebilir. Bunlar özellik uzayının kaynak ve hedef veri kümeleri için farklı olması ve aynı olması durumlarıdır. Örneğin bir mobil operatörün bir zamanlar yaptığı bir kampanyayı tekrar yapmak istediğinde eski kampanyanın bilgilerini kullanması. Bu problemde eğer kullanıcıların bilgileri tamamen aynı kalırsa özellik uzayı aynı olur, farklı bilgiler de eklenirse özellik uzayı farklı olur.

#### Gözetmensiz Öğrenme Aktarımı (Unsupervised Transfer Learning)

Tümevarımsal öğrenme aktarımına benzemektedir. Farklı olarak etiketsiz görevler üzerine odaklanmaktadır. Buradaki kaynak ve hedef veri kümeleri benzer fakat görevler birbirinden farklıdır. Kümeleme, boyut küçültme vb. de kullanılır.

### Transfer Öğrenmenin Kullanım Alanları

***Görüntü Sınıflandırma:***

Önceden ImageNet üzerinde eğitilmiş bir model, yeni bir veri kümesine uyarlanabilir. Örneğin; Röntgen görüntülerinden hastalık teşhisi.

***Nesne Algılama:***

Araçların algılanması ve sınıflandırılması

***Yüz Tanıma:***

Güvenlik sistemlerinde, önceden eğitilmiş bir CNN modeli kullanılarak yüz tanıma yapılır.

***Duygu Analizi:***

Müşteri yorumlarının pozitif/negatif olarak sınıflandırılması (ör. restoran incelemeleri).

***Makine Çevirisi:***

İngilizce'den Türkçe'ye çeviri yapmak için önceden eğitilmiş bir dil modeli kullanılabilir. Örneğin: Google Translate.

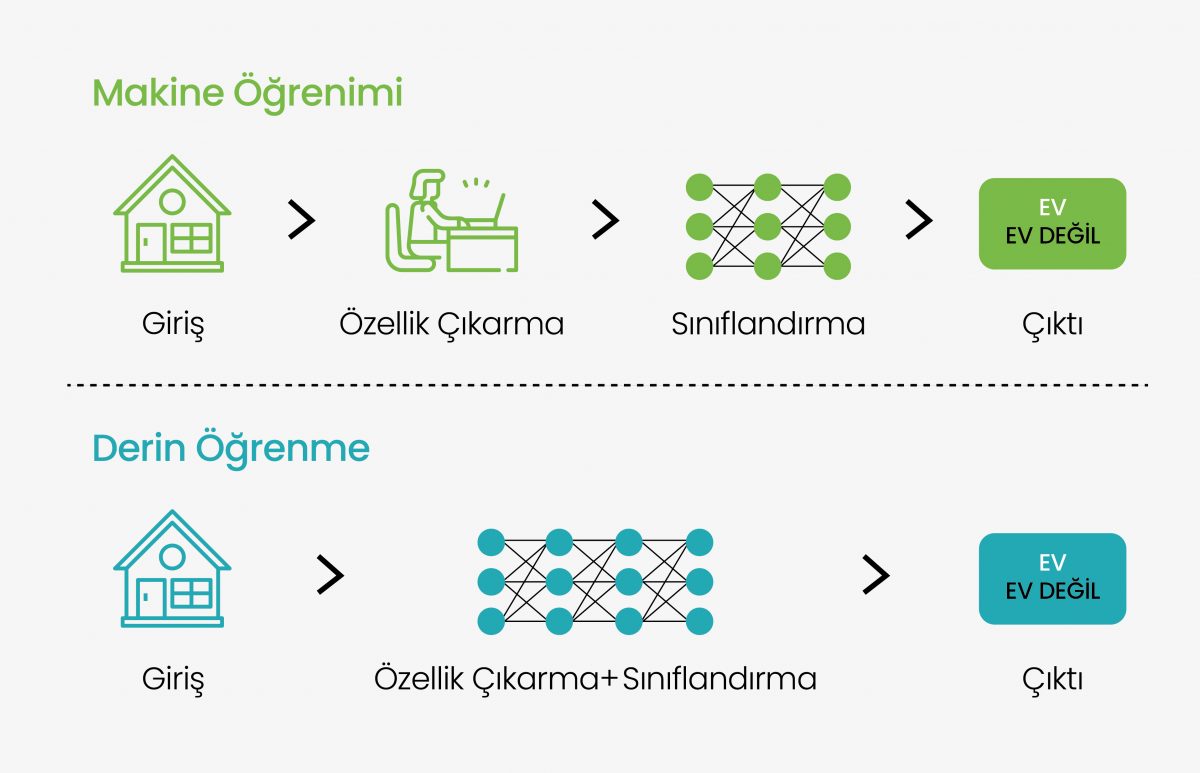
***Müzik Analizi:***

Önceden eğitilmiş bir model kullanılarak müzik türlerini sınıflandırma veya yeni melodi üretme.

***Öğrenci Performansı Tahmini:***

Öğrencilerin gelecekteki başarısını tahmin etmek için transfer öğrenme modelleri kullanılabilir.

## Makine Öğrenimi ve Derin Öğrenme Arasındaki Fark Nedir?



**Şekil 29: Makine Öğrenmesi ve Derin Öğrenme Arasındaki Fark**

**Makine Öğrenimi**

* Özellik çıkarımı için manuel mühendislik gereklidir.
* Daha az veriyle çalışabilir.
* Daha az hesaplama gücü gerektirir.
* Algoritmalar daha basittir.

**Derin Öğrenme**

* Özellikleri otomatik olarak öğrenir.
* Büyük veri setlerine ihtiyaç duyar.
* Yüksek hesaplama gücü (GPU/TPU) gerektirir.
* Çok katmanlı ve karmaşık yapılar kullanır.

## Veri Seti Nedir?

Veri seti, belirli bir probleme çözüm bulmak, bir süreç hakkında bilgi edinmek ya da bir makine öğrenmesi modelini eğitmek amacıyla toplanmış verilerden oluşur. Veri setleri, sistematik bir şekilde düzenlenmiş bilgi koleksiyonlarıdır ve makine öğrenmesi sürecinin temel yapı taşlarından biridir. Her veri seti, bir modelin öğrenme yeteneğini geliştirmek, doğrulamak ve test etmek için kritik bir rol oynar.

Veri setleri, genellikle satırlar ve sütunlardan oluşan tablolar şeklinde düzenlenir. Satırlar, bireysel gözlemleri ya da veri noktalarını temsil ederken, sütunlar her bir gözlem için farklı özellikleri (nitelikler ya da değişkenler) içerir. Ancak, veri setleri bu formatla sınırlı değildir; bazı veri setleri daha karmaşık hiyerarşik yapılandırmalar (örneğin JSON) ya da etiketlenmiş veri grupları (örneğin görüntüler) şeklinde de bulunabilir.

## Veri Seti Türleri

Veri setleri, verinin toplanma biçimine ve kullanım amacına göre farklı türlere ayrılabilir. Bu türler, verilerin analizinde kullanılan yöntemler ve stratejiler üzerinde doğrudan etkili olur. Aşağıda veri seti türlerini daha detaylı bir şekilde bulunmaktadır.

### *1. Eğitim Verisi (Training Dataset)*

Eğitim veri seti, modelin öğrenmesi için kullanılan veridir. Bu veri seti, modelin parametrelerini (örneğin ağırlıklar, yüksekler gibi) ayarlarken kullanılır. Model, bu veriler üzerinden örüntüleri öğrenir ve doğru tahminler yapabilmek için kendisini optimize eder.

### *2. Doğrulama Verisi (Validation Dataset)*

Doğrulama veri seti, modelin eğitim sürecindeki hatalarını belirlemek ve modelin genelleme yeteneğini test etmek için kullanılır. Eğitim sırasında, modelin çıktıları doğrulama verisi üzerinde değerlendirilir. Bu veri seti, modelin daha fazla öğrenme ve iyileştirme yapabilmesi için sürekli olarak güncellenir.

### *3. Test Verisi (Test Dataset)*

Test veri seti, modelin nihai doğruluğunu test etmek için kullanılır. Eğitim ve doğrulama süreçlerinin tamamlanmasının ardından, test verisi kullanılarak modelin gerçek dünya verilerine nasıl tepki vereceği ölçülür. Bu veri seti, modelin başarısını objektif bir şekilde değerlendirme imkânı sağlar.

### *4. Yapılandırılmış ve Yapılandırılmamış Veri*

Veri setleri, yapılandırılmış ve yapılandırılmamış olarak iki ana gruba ayrılabilir. Yapılandırılmış veri, satırlar ve sütunlar halinde düzenlenmiş, kolayca işlenebilen veri tiplerini içerirken, yapılandırılmamış veri, metin, video ve ses gibi düzensiz formatlarda bulunan verileri kapsar.

### *5. Zaman Serisi Verisi*

Zaman serisi verisi, belirli bir zaman dilimi içinde toplanan verilerdir. Bu veri setleri, özellikle finans, hava durumu gibi alanlarda yaygındır ve zamanla değişen trendleri analiz etmek için kullanılır.

### *6. Kategorik ve Sayısal Veri*

Veri setleri, kategorik ve sayısal veriler olmak üzere iki ana türde sınıflandırılabilir. Kategorik veriler, sınıflandırma içerir (örneğin, "Kadın" veya "Erkek") ve bu veriler üzerinde matematiksel işlemler yapılmaz. Sayısal veriler ise ölçülebilen ve genellikle matematiksel işlemler yapılabilen verilerdir (örneğin, bir kişinin yaşı, maaşı).

### 7. Büyük Veri Setleri (Big Data)

Büyük veri setleri, hacmi, hızı ve çeşitliliği bakımından geleneksel veri işleme araçlarının ötesine geçen veri setleridir. Bu veri türleri genellikle Hadoop veya Spark gibi dağıtık sistemlerde işlenir. Büyük veriler, sosyal medya, IoT cihazları, online alışveriş platformları gibi kaynaklardan gelir ve analiz edilmesi zaman alıcı ve karmaşık olabilir​.

Bu veri seti türleri, makine öğrenmesi ve veri bilimi projelerinde farklı teknikler ve algoritmalarla analiz edilir. Her tür veri setinin kendine özgü işleme ve analiz gereksinimleri vardır ve doğru veri setinin seçilmesi, modelin başarısını doğrudan etkiler.

## Veri Kümesinin Temel Analitik Özellikleri

Veri kümesinin analitik özellikleri, veri setinin genel yapısını, içindeki veri noktalarının dağılımını ve bu verilerin birbirleriyle olan ilişkilerini anlamada kritik rol oynar. Bu özellikler, özellikle veri analizi ve modelleme süreçlerinde önemli bir temel sağlar. Aşağıda, veri kümesinin temel analitik özellikleri detaylı olarak açıklanacaktır.

### *1. Veri Merkezi (Central Tendency)*

Veri merkezi, verilerin hangi noktada yoğunlaştığını gösteren bir özelliktir. Verilerin merkezi eğilimini ölçmek için genellikle ortalama, medyan ve mod gibi istatistiksel yöntemler kullanılır. Ortalama, tüm veri noktalarının toplamının, veri sayısına bölünmesiyle hesaplanır ve verilerin genel eğilimini yansıtır. Medyan, sıralanan veri kümesinin ortasında yer alan değeri temsil ederken, mod ise en fazla tekrar eden veri değerini ifade eder. Bu ölçütler, veri kümelerinin genel yapısını analiz etmek için önemli göstergelerdir.

### *2. Verilerin Çarpıklığı (Skewness)*

Veri çarpıklığı, veri dağılımının simetrik olup olmadığını gösterir. Çarpıklık değeri, verilerin sağa veya sola kayıp kaymadığını belirlemek için kullanılır. Pozitif çarpıklık, verilerin sağa doğru kaydığı ve daha büyük değerlerin yoğunlaştığı anlamına gelirken, negatif çarpıklık, verilerin sola doğru kayarak daha küçük değerlerin yoğunlaştığını belirtir. Çarpıklık analizi, veri setindeki simetrik olmayan dağılımları ve olası anormallikleri belirlemek için önemlidir.

### *3. Veri Yayılımı (Dispersion)*

Veri yayılımı, veri setinin merkezine göre ne kadar yayıldığını gösterir. Varyans ve standart sapma gibi ölçütler, verilerin yayılımını anlamada kullanılır. Varyans, veri noktalarının ortalamadan ne kadar sapma gösterdiğini ölçerken, standart sapma bu sapmaların ortalamasını ifade eder. Bu ölçütler, veri setindeki değerlerin çeşitliliğini ve çeşitlilikle ilgili olasılıkları daha net bir şekilde gösterir. Büyük bir varyans, veri noktalarının geniş bir aralığa yayıldığını, küçük bir varyans ise verilerin birbirine yakın olduğunu gösterir.

### *4. Aykırı Değerler (Outliers)*

Aykırı değerler, veri kümesinin genel eğiliminden sapmış ve olağan dışı değerlerdir. Bu tür değerler, genellikle modelleme süreçlerinde dikkate alınması gereken kritik faktörlerdir. Aykırı değerler, verilerin anlamlı bir şekilde işlenmesini engelleyebilir ve analiz sonuçlarını yanıltabilir. Aykırı değerlerin tespit edilmesi ve doğru bir şekilde işlenmesi, veri kalitesinin iyileştirilmesine yardımcı olur.

### *5. Korelasyon (Correlation)*

Korelasyon, iki ya da daha fazla değişken arasındaki ilişkinin gücünü ve yönünü ifade eder. Pozitif korelasyon, bir değişkenin artışıyla diğerinin de artış gösterdiğini belirtirken, negatif korelasyon bir değişkenin artışıyla diğerinin azaldığını gösterir. Korelasyon analizi, veri kümesindeki değişkenler arasındaki ilişkiyi anlamak ve bu ilişkileri modelleme sürecinde kullanmak için kritik öneme sahiptir .

## Veri Kümesinin Yapısal ve Format Özellikleri

Veri kümesinin yapısal ve format özellikleri, veri kümesinin içeriği, düzeni ve hangi biçimlerde saklandığını belirler. Bu özellikler, veri kümesinin analiz için uygunluğunu ve işlemeye olanak tanıyacak şekilde yapılandırılmasını sağlar.

### *1. Sayısal Özellikler (Numerical Features)*

Sayısal veriler, genellikle sürekli ya da kesikli ölçümlerle temsil edilen verilerdir. Bu veriler, istatistiksel işlemler yapılabilen verilerdir ve genellikle regresyon analizi ve benzeri matematiksel modellerde kullanılır. Örneğin, bir kişinin yaşı veya maaşı gibi sayısal veriler, modelleme sürecinde önemli girdi faktörleridir.

### *2. Kategorik Özellikler (Categorical Features)*

Kategorik veriler, belirli kategorilere veya sınıflara ayrılmış verilerdir ve genellikle metinsel ya da sınıf etiketlerine sahip olurlar. Cinsiyet, şehir veya ürün kategorileri gibi örnekler, kategorik verilere örnek teşkil eder. Kategorik veriler, sınıflandırma algoritmalarında kullanılır ve sayısal işlemler yapılamaz. Bu nedenle, kategorik verilerin uygun şekilde dönüştürülmesi ve işlenmesi gerekir.

### *3. Meta Veriler (Metadata)*

Meta veriler, bir veri kümesinin içeriğini ve yapısını açıklayan verilerdir. Büyük veri kümeleri, genellikle geniş ve karmaşık olduğundan, meta veriler, verilerin anlamını ve kullanılabilirliğini tanımlar. Meta veriler, veri setinin doğru bir şekilde anlaşılmasını sağlar ve veri kümesinin doğru yorumlanmasında yardımcı olur.

### *4. Veri Boyutu (Data Size)*

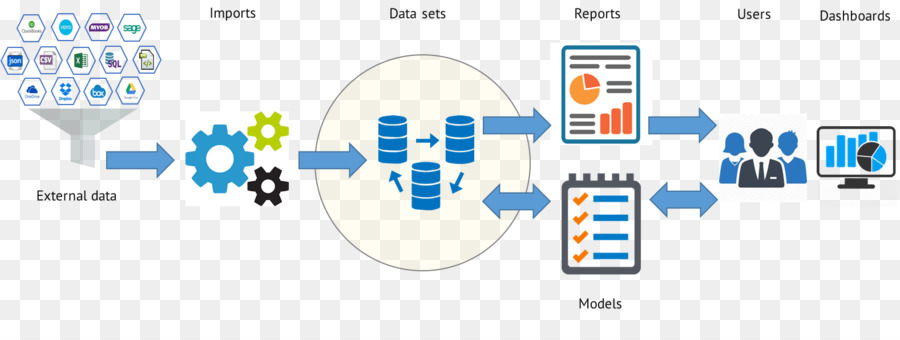
Veri boyutu, veri kümesinde bulunan toplam veri noktalarının sayısını ifade eder. Büyük veri kümeleri, genellikle özel depolama ve işleme gereksinimleri doğurur. Veri boyutunun büyük olması, analiz sürecinde hesaplama gücü ve depolama kapasitesinin artırılması gerektiği anlamına gelir.

### *5.Veri Biçimlendirilmesi (Data Formatting)*

Veri kümeleri, farklı biçimlerde sunulabilir. En yaygın veri formatları arasında CSV (Comma Separated Values), JSON (JavaScript Object Notation), XML (Extensible Markup Language) ve Excel dosyaları bulunur. Bu formatlar, verilerin düzenlenmesi ve işlenmesi sürecinde kullanılır ve veri kümesinin erişilebilirliğini sağlar.

### *6. Hedef Değişken (Target Variable)*

Makine öğrenmesi ve veri madenciliği süreçlerinde, hedef değişken, modelin tahmin etmeye çalıştığı çıktıdır. Bu değişken, genellikle bir sınıflandırma ya da regresyon modeli ile tahmin edilmek üzere seçilir. Hedef değişkenin doğru seçilmesi, modelin başarısını doğrudan etkiler.



**Şekil 30**

## Veri Setlerinde Kullanılan Yöntemler

Veri kümeleriyle çalışırken birçok yöntem uygulanır. Bu, verilen veri kümenizle çalışma nedeninize bağlıdır. Veri kümelerine uygulanan yaygın yöntemlerden bazıları şunlardır:

### *1. Veri Kümelerini Yükleme ve Okuma:*

Gerekli işlemlerin yapılması için başlangıçta veri kümelerinin okunmasında ve okunmasında kullanılan bilgisayarların kurulumu yapılır.

Örneğin; read\_csv(), read\_json(), read\_excel() vb.

### *2. Keşifsel Veri Analizi:*

Veri seti yapmak ve görselleştirmek için , bu fonksiyonları bir veri kümesi üzerinde çalışmak üzere kullanırız.

Örneğin; kafa(), kuyruk(), groupby() vb.

### *3. Veri Ön İşleme:*

Bir veri kümesini analiz etmeden önce, parçalar halinde kırılan hatalı değerler ve yanlış etiketlenmiş veri noktalarının çıkması için dağıtılır.

Örneğin; drop(), fillna(), dropna(), copy() vb.

### *4. Veri Manipülasyonu:*

Verilerin ayrılmasındaki veri noktalarını, özellikleri manipüle etmek için ürünleri/yeniden ürünleri sunar. Bazı noktalarda, veri kurulumunun özellikleri bile hesaplama karmaşıklığını en aza indirmek için manipüle edilir ve bu şekilde devam eder. Bu, sütunları sınırlama, yeni veri bölümleri ekleme vb. yöntemleri arasında dağıtma.

Örneğin; kırılma (), kırılma (), kırılma () vb.

### *5. Görselleştirmeyi Verin:*

Veri ayarı, teknik alanda olmayan kişiler için kullanılan kopyalar, örneğin: şirket/işletmenin veri ayarının resimli bir temsilini sağlamak için çubuk grafiklerin ve parlaklıkların kullanılması.

Örneğin; arsa()

### *6. Veri İndeksleme, Veri Alt Kümeleri:*

Bir veri ayarındaki belirli bir parçanın mevcut olması için kullanılan işlemler, veri indekslemeyi kullanır veya kesin alt kümeler oluşturulur.

Örneğin; iloc()

### *7. Dış Veri Aktarımı:*

Üzerindeki verilerin verileri gibi farklı biçimlerde aktarmak için kullanılan yöntemler.

Örneğin; to\_csv(), to\_json() vb.

## Veri Setlerinin Kullanım Alanları

Veri setleri, pek çok farklı alanda kullanılan ve analizler yapmaya yarayan önemli araçlardır. Özellikle makine öğrenmesi, veri madenciliği, yapay zekâ ve istatistiksel analiz gibi disiplinlerde temel rol oynarlar. Veri setlerinin kullanım alanları, sadece akademik araştırmalarla sınırlı kalmaz; sağlık, ekonomi, çevre bilimleri gibi çeşitli alanlarda da önemli uygulamaları bulunmaktadır.

### *1. Makine Öğrenmesi ve Yapay Zekâ*

Makine öğrenmesi ve yapay zekâ alanlarında, veri setleri temel yapı taşlarıdır. Model eğitimi için kullanılan veri setleri, genellikle etiketli (supervised learning) veya etiketsiz (unsupervised learning) şekilde sınıflandırılır. Örneğin, MNIST veri seti, el yazısı rakamları tanıyan modellerin eğitiminde yaygın olarak kullanılmaktadır. Bu tür veri setleri, modelin öğrenmesini ve tahmin yapmasını mümkün kılar

### *2. Sağlık Bilimleri*

Sağlık verileri, hastalıkların izlenmesi, tedavi yöntemlerinin geliştirilmesi ve sağlık politikalarının şekillendirilmesi için kullanılır. CDC (Centers for Disease Control and Prevention) gibi kurumlar, sağlık verilerini toplayarak araştırmacılara, doktorlara ve sağlık politikası oluşturucularına kararlar alırken rehberlik eder. Örneğin, sağlıkla ilgili veri setleri, bulaşıcı hastalıkların yayılımı gibi bilgileri içerebilir. Bu tür veriler, halk sağlığına yönelik önlemlerin geliştirilmesinde büyük rol oynar.

### *3. Çevre Bilimleri*

Çevre verileri, çevresel değişikliklerin izlenmesi ve ekolojik dengeyi koruma çalışmalarında önemli bir yer tutar. Hava kirliliği, su kaynaklarının kirlenmesi, orman alanlarının tahrip edilmesi gibi çevresel faktörler, düzenli veri setleriyle izlenebilir. Örneğin, New York City’nin hava kalitesi veri seti, şehirdeki hava kirliliği seviyelerini analiz etmek için kullanılmaktadır. Bu veriler, çevre politikalarının oluşturulmasında ve hava kalitesinin iyileştirilmesine yönelik stratejilerin belirlenmesinde kritik öneme sahiptir.

### *4. Ekonomik ve Finansal Analizler*

Veri setleri, ekonomik ve finansal verilerin analiz edilmesinde de sıklıkla kullanılır. Federal Depozito Sigorta Kurumu (FDIC) gibi finansal denetim kurumları, iflas eden bankalarla ilgili veri setleri sunarak finansal sistemin analiz edilmesine katkıda bulunurlar. Bu veri setleri, bankacılık sektöründeki krizlerin önlenmesine, ekonomik büyümenin izlenmesine ve finansal risklerin yönetilmesine yardımcı olur. Ayrıca, hükümet politikalarının yönlendirilmesinde de büyük rol oynar.

### *5. Sosyal Bilimler*

Sosyal bilimlerde, veri setleri toplumsal eğilimleri analiz etmek, demografik değişimleri gözlemlemek ve toplumsal problemleri incelemek için kullanılır. Bu veriler, genellikle anketler, nüfus sayımları ve kamuya açık veriler gibi kaynaklardan elde edilir. Örneğin, ABD Nüfus Sayımı, sosyal bilimcilerin nüfusun demografik özelliklerini incelemesine olanak tanırken, kamu politikalarının şekillendirilmesinde de kullanılmaktadır.

## Değerlendirme Metrikleri

Makine öğrenimi modellerinin performansını değerlendirmek için çeşitli değerlendirme metrikleri kullanılır. Bu metrikler, modelin başarı düzeyini ölçmek ve farklı modelleri karşılaştırmak için kullanılır. Hangi metriğin kullanılacağı, probleme (örneğin, sınıflandırma, regresyon) ve problemdeki veri dengesine bağlıdır.

### *1. Doğruluk (Accuracy)*

Doğruluk, modelin doğru tahminlerinin toplam tahminlere oranıdır. Sınıflandırma problemleri için yaygın olarak kullanılan bir metriktir.

Doğruluk =



**Örnek:**

100 veriden 90'ını doğru tahmin ettiyse:

Doğruluk = = 0.90 (90%)



**Avantajları:**

Veri dengeli olduğunda anlamlıdır.

**Dezavantajları:**

Veri sınıfları dengesiz olduğunda yanıltıcı olabilir (örneğin, %95 bir sınıf, %5 diğer sınıf).

### *2. Hata Oranı (Error Rate)*

Hata oranı, modelin yanlış tahmin ettiği oranı ifade eder.

Hata oranı = 1− Doğruluk

**Örnek:**

Doğruluk 0.90 ise, hata oranı:

Hata oranı = 1−0.90=0.10 (10%)

**Kullanım:**

Doğrulukla ters ilişkilidir. Hata oranı düşükse model başarılıdır.

### *3. F1 Skoru (F1 Score)*

F1 skoru, **Precision** (Kesinlik) ve **Recall** (Duyarlılık) metriklerinin harmonik ortalamasıdır. Dengeli bir metrik olduğu için özellikle dengesiz veri kümelerinde kullanışlıdır.

Fı skoru = 2.



**Precision (Kesinlik):** Modelin pozitif olarak tahmin ettiği verilerden kaçı doğru

Kesinlik =



**Recall (Duyarlılık):** Gerçek pozitiflerin ne kadarını doğru tahmin etti.

Kesinlik =



**Avantajı:**

Precision ve Recall arasında bir denge kurar.

## MAKİNE ÖĞRENMESİNDE KARŞILAŞILAN SORUNLAR

### AŞIRI UYUM (OVERFIT)

Overfitting, makine öğrenmesinde bir modelin eğitim verilerine aşırı derecede uyum sağlaması durumunu ifade eder. Bu durumda, model eğitim verilerindeki gürültüyü, rastgele varyasyonları ya da örnek veri noktalarındaki özel özellikleri öğrenir. Sonuç olarak, model, eğitim verilerine mükemmel bir şekilde uymakla kalmaz, aynı zamanda yeni ve görünmemiş veriyle karşılaştığında hatalı veya yanıltıcı tahminlerde bulunabilir.

Genellikle eğitim verilerinde düşük hata elde ederken, doğrulama verilerinde daha yüksek bir hata ile kendini gösterir. Model eğitim verilerine çok fazla uyum sağladığında, bu verilerle mükemmel bir performans gösterir, ancak yeni verilerle başa çıkmakta zorlanır.

Aşırı uyum sorununu önleyebilmek için yapılabilecek stratejiler:

Eğitim verileri artırılabilir.

Model karmaşıklığını azaltılabilir.

Birbiri ile korelasyon içinde olan sütunlar silinebilir.

Eğitim ve test verilerinde hatalar arasındaki fark belli bir seviyedeyken eğitim durdurulabilir.

### EKSİK UYUM (UNDERFIT)

Aşırı öğrenmenin aksine, bir model yetersiz öğrenmeye sahipse, modelin eğitim verilerine uymadığı ve bu nedenle verilerdeki trendleri kaçırdığı anlamına gelir. Ayrıca modelin yeni veriler için genelleştirilemediği anlamına da gelir.

Underfitting sorunu olan modellerde hem eğitim hem de test veri setinde hata oranı yüksektir. Düşük varyans ve yüksek bias’a sahiptir. Bu modeller eğitim verilerini çok yakından takip etmek yerine, eğitim verilerinden alınan dersleri yok sayar ve girdiler ile çıktılar arasındaki temel ilişkiyi öğrenemez.

Eksik uyum sorununu önleyebilmek için yapılabilecek stratejiler:

Model karmaşıklığı arttırılabilir.

Özelliklerin sayısı arttırılabilir.

### VERİ DENGESİZLİĞİ

Sınıflandırma yaparken, sınıfların eşit dağılmadığı, yani her sınıf için yaklaşık olarak aynı sayıda verinin olmadığı veri kümeleri bulunduğunda bu duruma veri dengesizliği adı verilir. Örneğin: ikili sınıflandırma (binary classification) durumunda, 500 verinin olduğu bir veri kümesindeki 40 verinin azınlık sınıfa (Sınıf-1), 460 verinin ise diğer sınıfa (Sınıf-0) ait olması durumudur.

Dengesiz veri kümeleri ile çalıştığımızda elde edilen sınıflandırıcılar (classifiers), genel hata oranını minimize etmeye çalışırken ağırlıklı olan sınıfı (Sınıf-0) kategorize ederek, azınlık sınıfını (Sınıf-1) göz ardı edebiliyor. Bu durumlarda asıl hedef genellikle azınlık olan sınıfı (Sınıf-1) doğru sınıflandırabilmek olduğu için, dengesiz veri kümeleriyle çalışırken şunlara dikkat etmeliyiz:

Doğru metrik seçimi

Katmanlama (Stratification)

Yeniden Örnekleme (Resampling)

Sınıf Ağırlıklarını Ayarlamak (Adjusting Class Weights)

Penalized Modeller

Anomali Tespiti (Detection)

### HATALI ÖZELLİK MÜHENDİSLİĞİ

Özellik mühendisliği, makine öğrenmesinde modelin öğrenmesi için en uygun veri temsillerinin oluşturulması sürecidir. Ancak yanlış veya yetersiz özellik mühendisliği, modelin performansını ciddi şekilde etkileyebilir. Bu durum, modelin verilerdeki temel ilişkileri öğrenememesine ve kötü genelleştirme yapmasına yol açar.

Hatalı özellik mühendisliği genellikle şu şekillerde ortaya çıkar:

***Gereksiz ya da alakasız özellikler:*** Modelin öğrenmesini zorlaştırarak gürültü ekler ve aşırı uyuma sebep olabilir.

***Eksik veya hatalı özellik seçimi****:* Model için anlamlı olan özelliklerin belirlenmemesi durumunda, model önemli ilişkileri gözden kaçırır.

***Özelliklerin doğru ölçeklendirilmemesi:*** Bazı algoritmalar (örneğin, destek vektör makineleri ve k-en yakın komşu) özellikle verilerin ölçeğine duyarlıdır.

***Aykırı değerlerin göz ardı edilmesi:*** Önemli özelliklerdeki uç değerlerin dikkate alınmaması modelin doğruluğunu düşürebilir.

Hatalı özellik mühendisliğini önleyebilmek için yapılabilecek stratejiler:

Veriyi analiz ederek gereksiz ve alakasız özellikleri çıkarmak.

Özelliklerin ölçeklendirme (normalizasyon veya standardizasyon) işlemlerini uygulamak.

Özellik seçimi ve boyut indirgeme tekniklerini kullanmak (Örneğin: PCA, LASSO).

Aykırı değerleri tespit edip uygun bir şekilde işlemek.

Domain bilgisine dayalı yeni özellikler oluşturmak.

### MODEL SEÇİMİ VE HİPERPARAMETRE OPTİMİZASYONU

Makine öğrenmesi projelerinde, doğru modelin seçilmesi ve hiperparametrelerin optimize edilmesi, model performansı açısından kritik bir rol oynar. Yanlış model seçimi veya optimal olmayan hiperparametreler, düşük doğruluk, yüksek hata oranları ve genelleştirme problemlerine neden olabilir.

**Model Seçimi:**

Model seçimi, problem türüne (regresyon, sınıflandırma vb.), veri büyüklüğüne ve veri yapısına bağlı olarak yapılmalıdır.

Karmaşık problemler için daha sofistike modeller (örneğin, karar ağaçları, derin öğrenme) kullanılırken;

Küçük veri kümelerinde daha basit modeller (örneğin, lineer regresyon veya lojistik regresyon) tercih edilebilir.

**Hiperparametre Optimizasyonu:**

Modeller, veri üzerinde eğitilirken bazı ayarlanması gereken hiperparametrelerle gelir. Yanlış hiperparametreler modelin başarısız olmasına neden olabilir.

Hiperparametre optimizasyonunda yaygın yöntemler:

***Grid Search:***Belirli hiperparametre aralıkları taranarak en uygun kombinasyon bulunur.

***Random Search:*** Rastgele hiperparametre kombinasyonlarını test ederek zaman tasarrufu sağlar.

***Bayesian Optimization:*** Daha akıllı bir şekilde hiperparametre uzayını araştırır.

Hiperparametre optimizasyonunun önemi:

Modelin genelleştirme yeteneğini iyileştirir.

Eğitim süresini ve hesaplama yükünü optimize eder.

Çözüm Önerileri:

Verinin türüne ve büyüklüğüne uygun modeli seçmek.

Kıyaslama (benchmark) yöntemleriyle farklı modelleri karşılaştırmak.

Doğru hiperparametre optimizasyon tekniklerini kullanarak modeli iyileştirmek.

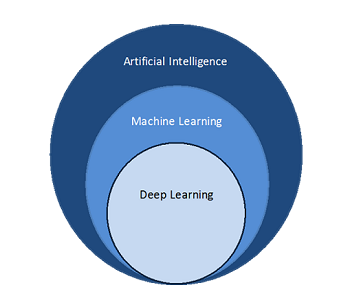
Çapraz doğrulama (cross-validation) yöntemiyle modelin performansını değerlendirmek.

## MAKİNE ÖĞRENMESİNİN DİĞER ALANLARLA FARKLARI VE İLİŞKİLERİ

Makine öğrenmesinin diğer alanlarla farkları ve ilişkileri oldukça çeşitlidir. Bu ilişkiler her alanın kendi metodolojik çerçeveleriyle şekillenirken makine öğrenmesinin veri odaklı ve öğrenmeye dayalı yaklaşımı her bir disiplinde kendine özgü çözümler sunar. Makine öğrenmesinin avantajları her sektörde analitik süreçlerin hızlanmasını ve daha doğru tahminlerin yapılmasını sağlarken diğer disiplinlerle olan etkileşimleri karmaşık problemleri çözmede önemli bir rol oynamaktadır. İşte bazı önemli alanlar:

### *Yapay Zekâ* *(AI) ve Makine Öğrenmesi*

Makine öğrenmesi, yapay zekânın (AI) bir alt alanıdır. Yapay zekâ , makinelerin insan benzeri düşünme, öğrenme ve karar verme yeteneklerine sahip olmasını sağlamak için daha geniş bir çerçevede çalışırken makine öğrenmesi özellikle verilerden öğrenme ve tahminler yapma konusunda odaklanır. Yani AI daha genel bir kavramken, makine öğrenmesi bu kavramın bir uygulama alanıdır.



### *Veri Bilimi ve Makine Öğrenmesi*

Veri bilimi, verileri analiz etme ve anlamlandırma sürecini kapsar, makine öğrenmesi ise bu süreçte veriden öğrenmeye yönelik teknikleri kullanır. Veri bilimi temizleme, görselleştirme ve çıkarım yapma gibi süreçleri içerirken, makine öğrenmesi daha çok tahmin yapma ve model oluşturma üzerine yoğunlaşır. Veri bilimi, makine öğrenmesi algoritmalarını uygulamak ve sonuçları yorumlamak için altyapı sağlar.

**Şekil 31: Yapay Zekâ** **- Makine Öğrenmesi - Derin Öğrenme İlişkisi**

### *Derin Öğrenme ve Makine Öğrenmesi*

Derin öğrenme, makine öğrenmesinin bir alt kümesidir. Derin öğrenme daha karmaşık ve çok katmanlı yapay sinir ağları kullanarak büyük veri setlerinde daha derin özellikler öğrenir. Makine öğrenmesi ise daha basit algoritmalarla çalışarak daha küçük veri setlerinde hızlı ve verimli sonuçlar üretir. Derin öğrenme büyük veri ve işlem gücü gerektirirken, makine öğrenmesi genellikle daha az kaynakla çalışabilir. Derin öğrenme görüntü tanıma ve ses işleme gibi karmaşık görevlerde daha başarılı olurken, makine öğrenmesi basit tahmin ve sınıflandırma işlemleri için idealdir.

### *İstatistik ve Makine Öğrenmesi*

İstatistiksel yöntemler, veri analizi ve çıkarımlar için geleneksel bir yaklaşımdır. Makine öğrenmesi, istatistiksel modelleme temellerine dayansa da daha büyük ve karmaşık veri setlerinde daha güçlü sonuçlar elde etmek için algoritmalar kullanır. İstatistik, genellikle parametrelerin belirlenmesine dayanırken, makine öğrenmesi veri setlerinde otomatik öğrenme üzerine yoğunlaşır.

### *Bilgisayarla Görüntü İşleme ve Makine Öğrenmesi*

Görüntü işleme, bilgisayarların görsel verileri analiz etmesini sağlayan bir alandır. Makine öğrenmesi ise bu verileri analiz etme ve sınıflandırma gibi görevlerde kullanılabilir. Özellikle derin öğrenme (Deep Learning), görüntü işleme alanında oldukça etkili olup otomatik etiketleme, nesne tanıma ve yüz tanıma gibi uygulamalar için sıklıkla kullanılır.

### *Doğal Dil İşleme (NLP) ve Makine Öğrenmesi*

NLP, bilgisayarların insan dilini anlamasını ve işlemesini sağlar. Makine öğrenmesi, NLP’de metin sınıflandırma, dil modelleme, sentiment analizi gibi görevlerde aktif bir rol oynar. Özellikle derin öğrenme yöntemleri, metin verilerinde daha karmaşık anlamları öğrenmek ve dilin inceliklerini analiz etmek için kullanılır.

### *Big Data ve Makine Öğrenmesi*

Big Data çok büyük miktarda ve çeşitli veri setlerini ifade ederken, makine öğrenmesi bu verilerden anlam çıkarma ve tahmin yapma süreçlerine odaklanır. Büyük veri, veri toplama ve işleme sürecini kapsar, makine öğrenmesi ise bu veriler üzerinde öğrenme ve model oluşturma tekniklerini kullanır. Temelde, büyük veri miktarına odaklanırken, makine öğrenmesi bu verileri analiz ederek anlamlı sonuçlar elde etmeyi amaçlar.

Makine öğrenmesinin bu alanlarla ilişkisi, her birinin kendi hedeflerine odaklanmasına rağmen, makine öğrenmesinin bu hedeflere ulaşmada veriye dayalı yöntemlerle daha verimli sonuçlar üretmesini sağlar. Her alan kendi metodolojisini kullanırken, makine öğrenmesi daha dinamik ve veri odaklı bir yaklaşım sunar. Makine öğrenmesi birçok disiplini etkileyen ve dönüştüren güçlü bir araç araçtır.

## MAKİNE ÖĞRENMESİNDEKİ RİSKLER

### *Önyargı ve Ayrımcılık:*

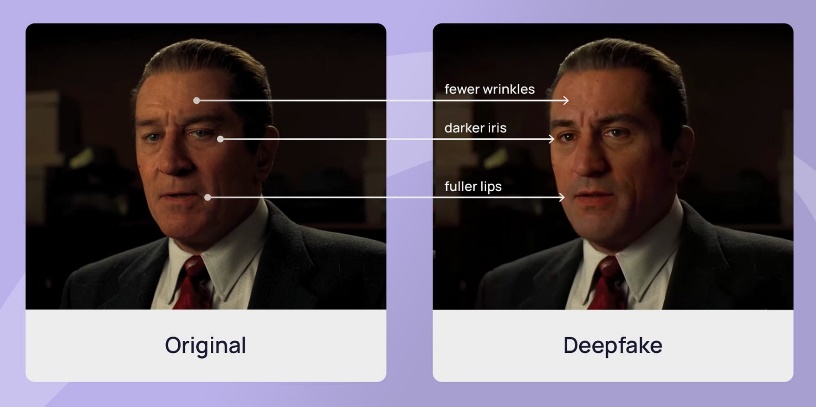
Makine öğrenmesi algoritmaları, eğitim verilerinde bulunan önyargıları öğrenebilir ve bu önyargıları kararlarına yansıtabilir. Bu durum, bazı grupların haksız yere dezavantajlı duruma düşmesine neden olabilir. Örneğin, bir işe alım algoritması geçmişte daha çok erkek adayların tercih edildiği bir veri setiyle eğitilmişse, kadın adayları sistematik olarak daha düşük değerlendirebilir veya dezavantajlı konuma düşürebilir.

### *Veri Gizliliği ve Güvenliği:*

Makine öğrenmesi sistemleri, genellikle büyük miktarda kişisel veriye dayanır. Bu gizlilik ihlalleri ve kötüye kullanım risklerini artırır. Verilerin sızması, bireylerin mahremiyetinin ihlaliyle sonuçlanabilir. Ayrıca, bu veriler yetkisiz erişimlere ve siber saldırılara karşı savunmasız olabilir.

### *Derin Sahtecilik (Deepfake):*

Makine öğrenmesi sahte görüntü, ses veya video (deepfake) üretiminde kullanılabilir. Bu durum, bireylerin itibarını zedeleyebilir, yanlış bilgilere dayalı toplumsal algı oluşturulmasına neden olabilir ve toplumsal güveni sarsabilir.

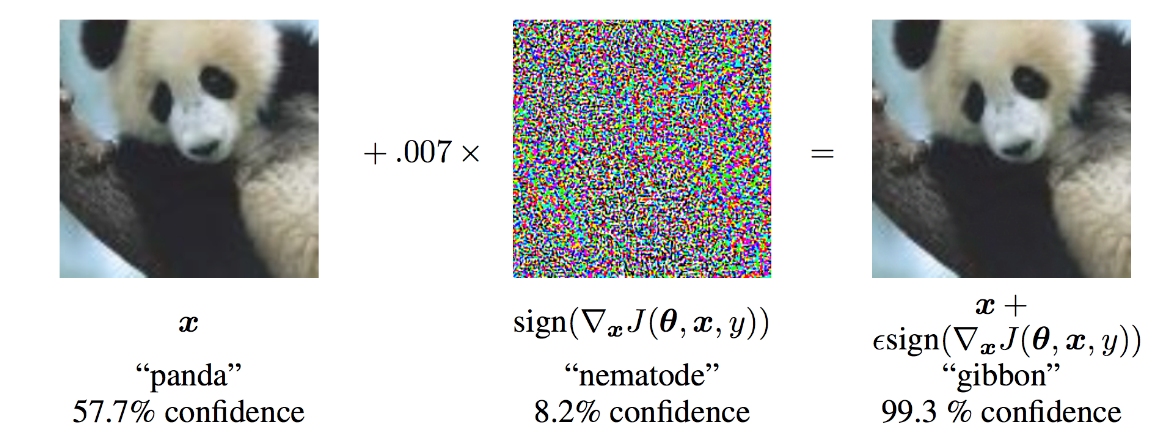


**Şekil 32: Derin Sahtecilik Örneği**

### *Adversarial Saldırılar:*

Makine öğrenmesi modelleri, özellikle düşmanca manipülasyonlara (adversarial saldırılar) karşı savunmasızdır . Adversarial saldırılar, makine öğrenmesi modellerini yanıltmak için genellikle insan gözüyle fark edilmeyen küçük değişiklikler yapar. Bu değişiklikler, matematiksel olarak modellenmiş ve algoritmaların öğrenme sürecini manipüle edecek şekilde hesaplanır. İnsan gözü bu değişiklikleri fark edemese de model bu değişiklikleri "yanlış" olarak algılar ve hatalı bir sonuç üretir. Örneğin, bir görüntü tanıma modeline yönelik bir adversarial saldırı, bir fotoğrafın piksel değerlerini o kadar ince bir şekilde değiştirir ki insanın gözünde hiçbir fark görülmez. Ancak bu küçük değişiklikler modelin kararını bozarak, bir kedi resmini yanlışlıkla köpek olarak tanımlamasına neden olabilir. Bu tür saldırılar özellikle güvenlik, sağlık ve otonom araçlar gibi kritik alanlarda ciddi riskler oluşturabilir.

Bunun bir örneği, aşağıdaki panda resmidir; insan gözünün tanıyamayacağı bir miktar parazit (gürültü) eklendiğinde, bir gibon resmi olarak sınıflandırılır.



**Şekil 33: Adversarial Saldırı Örneği**

### *Şeffaflık Eksikliği:*

Makine öğrenmesi modellerinin nasıl çalıştığı ve hangi kriterlere göre karar aldığı genellikle net bir şekilde anlaşılamaz. Modellerin karar mekanizmaları oldukça karmaşık olduğu için, kullanıcılar ve denetçiler bu süreçleri incelemekte zorlanabilir. "Kara kutu" olarak adlandırılan bu yapılar , hem bireyler hem de kurumlar için sorumluluk ve güven sorunları yaratabilir.

### *İş Kaybı ve Sosyoekonomik Etkiler:*

Otomasyon ve makine öğrenmesi uygulamaları, birçok sektörde insan gücüne olan ihtiyacı azaltarak iş kayıplarına yol açabilir. Özellikle düşük vasıflı işlerde çalışan bireyler bu dönüşümden olumsuz etkilenebilir. Bu durum, işsizliğin yanı sıra gelir eşitsizliğinin artmasına da neden olabilir.

### *Hatalı Kararlar:*

Makine öğrenmesi modelleri, eğitim verilerindeki yanlışlıklar veya hatalı yapılandırmalar nedeniyle hatalı kararlar alabilir. Örneğin, bir sağlık teşhis sisteminin yanlış teşhis koyması, ciddi sonuçlara yol açabilir.

### *Kötüye Kullanım Potansiyeli:*

Makine öğrenmesi, yanlış ellerde büyük bir tehdit oluşturabilir. Sahte bilgi yaymak, siber saldırıları daha karmaşık hale getirmek veya bireylerin izni olmadan gözetim yapmak gibi etik dışı amaçlarla kullanılabilir.

### *Regülasyon Eksikliği:*

Makine öğrenmesi sistemlerinin kullanımını düzenleyen yeterli yasal çerçevelerin bulunmaması, teknolojinin kontrolsüz bir şekilde yayılmasına yol açabilir. Bu durum, kötüye kullanım risklerini artırabilir ve toplumsal güvenin zedelenmesine neden olabilir.

### *İnsan Yetkinliklerinde Azalma:*

Makine öğrenmesi sistemlerinin aşırı kullanımı, insanların fiziksel ve düşünsel yetkinliklerini kaybetmesine yol açabilir. Sürekli olarak algoritmaların rehberliğine güvenmek, insanların karar verme, eleştirel düşünme ve problem çözme gibi zihinsel becerilerinin körelmesine neden olabilir. Benzer şekilde, fiziksel işleri otomatikleştiren sistemler, insanların kas hafızası ve el becerileri gibi yeteneklerinde gerilemeye sebep olabilir. Bu, bireylerin hem zihinsel hem de fiziksel olarak hamlaşmasına yol açar ve uzun vadede bireylerin teknolojiye olan bağımlılığını artırır.

## RİSKLERE KARŞI NASIL ÖNLEMLER ALINABİLİR?

Makine öğrenmesi teknolojilerinin sorumlu bir şekilde geliştirilmesi ve uygulanması, bu risklerin önüne geçmek için hayati öneme sahiptir.

Makine öğrenmesindeki risklere karşı alınacak önlemler, birkaç temel strateji etrafında şekillenir. İlk olarak, modelin eğitiminde kullanılan verilerin kalitesi büyük önem taşır. Verilerin doğru, güncel ve dengeli olması, modelin doğru sonuçlar üretmesini sağlar. Ayrıca, yanlılık (bias) riski en aza indirilmelidir; ırk, cinsiyet veya diğer demografik faktörlere dayalı ayrımcılığın önüne geçilmesi için veriler titizlikle incelenmelidir. Modelin şeffaflığını artırmak da kritik bir adımdır; açıklanabilir yapay zekâ yöntemleri ile kullanıcılar ve geliştiriciler, modelin nasıl ve neden belirli kararlar aldığını anlayabilmelidir. Sürekli izleme ve güncelleme, modelin zamanla değişen verilere uyum sağlamasına yardımcı olur. Ayrıca, devletler makine öğrenmesi alanında düzenleyici yasalar çıkararak, veri gizliliğini koruyabilir, siber güvenliği güçlendirebilir ve etik kurallara uyumu sağlayabilir. Otomatik karar verme sistemlerinin denetlenmesi ve insan müdahalesinin gerektiği durumların belirlenmesi de önemli bir önlem olarak öne çıkar. Son olarak, kullanıcı geri bildirimleri ve çeşitli disiplinlerden gelen uzman görüşleri ile model sürekli geliştirilerek, güvenli ve toplumsal fayda sağlayacak şekilde kullanılabilir. Bu önlemler, makine öğrenmesinin potansiyel zararlarını en aza indirirken, toplum için daha güvenli ve adil bir teknoloji ortamı oluşturur.

## MAKİNE ÖĞRENMESİNDE ETİK

Makine öğrenmesi etiği, yapay zekânın toplumda güvenli ve adil bir şekilde kullanılmasını sağlamak için kritik bir alandır. Bu alandaki temel konular, sistemlerin adil, güvenli ve şeffaf bir şekilde çalışmasını sağlamak için önemlidir.

*Önyargı ve Ayrımcılık:*  
Makine öğrenmesi modelleri, eğitildikleri verilerdeki önyargıları öğrenebilir ve bu da ayrımcılığa yol açabilir. Bu önyargıların önlenmesi için verilerin çeşitlendirilmesi ve algoritmaların dikkatli bir şekilde eğitilmesi gerekmektedir.

*Şeffaflık ve Hesap Verebilirlik:*  
Makine öğrenmesi modelleri genellikle karmaşık ve anlaşılmazdır. Bu nedenle sistemlerin şeffaf olması, kararlarının izlenebilir ve açıklanabilir olması gerekmektedir. Aksi takdirde, modelin neden bir karar verdiği sorgulanamaz ve sorumluluk sorunları ortaya çıkabilir.

*Gizlilik ve Veri Koruma:*  
Makine öğrenmesi algoritmaları, kişisel verileri toplar ve işler, bu da gizlilik ihlalleri riskini doğurur. Kullanıcıların verilerinin korunması ve yalnızca izin verilen şekilde kullanılmasının sağlanması kritik öneme sahiptir.

*Adalet ve Eşitlik:*  
Makine öğrenmesi algoritmaları, karar verme süreçlerinde adaletli ve eşit olmalıdır. Bu özellikle hukuk, sağlık, finans gibi sektörlerde büyük önem taşır. Algoritmalar ayrımcılığa yol açmadan, herkese eşit fırsatlar sunacak şekilde tasarlanmalıdır. Eğitim verilerindeki önyargıların algoritmalara yansıması engellenmeli, model çıktıları her bireyi adil bir şekilde etkilemelidir.

Makine öğrenmesi etiği, yalnızca teknolojinin doğru şekilde uygulanmasını sağlamaz, aynı zamanda bu teknolojilerin toplumun faydasına sunulmasına da katkı sağlar. Bu nedenle etik ilkeler doğrultusunda geliştirilen ve kullanılan makine öğrenmesi sistemleri, daha güvenli, adil ve sürdürülebilir bir geleceğe katkı sağlar.

## GÜNCEL GELİŞMELER VE TRENDLER

Makine öğrenmesindeki güncel gelişmeler ve trendler 2024 yılı itibarıyla bir dizi heyecan verici yenilik içeriyor:

### *Otomatik Makine Öğrenmesi (AutoML)*

Veri etiketleme ve model optimizasyonu gibi karmaşık süreçlerin otomatikleştirilmesi, daha hızlı ve verimli model geliştirmeyi mümkün kılıyor. Bu, daha geniş bir kullanıcı kitlesine, teknik bilgi gereksinimlerini en aza indirerek makine öğrenmesi uygulamaları sunuyor. Bu alanda iki örnek verilebilir: Google AutoML , H20.ai AutoML . Google AutoML , veri etiketleme ve model optimizasyonunu otomatikleştirerek, kullanıcıların minimum teknik bilgi ile güçlü yapay zekâ uygulamaları geliştirmelerini sağlıyor. H2O.ai, makine öğrenmesi için açık kaynaklı bir AutoML platformu sunuyor. Bu platform, kullanıcıların veriye dayalı uygulamalar oluştururken, veri ön işleme, model seçimi ve hiperparametre ayarlarını otomatikleştiren araçlar sunuyor.

### *Açıklanabilir Yapay Zekâ (XAI)*

Yapay zekâ sistemlerinin karar alma süreçlerinin daha anlaşılır hale gelmesi, özellikle kritik alanlarda, algoritmalara duyulan güveni artırıyor. Örneğin, sağlık ve finans gibi sektörlerde, kararların nasıl alındığının izlenebilir olması, kullanıcılar ve denetçiler için daha güvenli bir ortam sunuyor. Bu, etik ve adaletin sağlanmasına katkı sağlıyor. Bu alanda iki örnek verilebilir: IBM Watson, Google Cloud AI Explanations. IBM, XAI'yi geliştirmeye yönelik çalışmalara öncülük ediyor. Watson AI, kullanıcıların makine öğrenmesi model sonuçlarını daha iyi anlamalarına olanak tanır. Özellikle sağlık ve finans sektörlerinde, XAI ile karar alma süreçlerini daha şeffaf hale getirmeyi amaçlıyor. Google, AI ve makine öğrenmesi modellerinin açıklanabilirliğini artıran "AI Explanations" aracıyla kullanıcıların model sonuçlarını nasıl elde ettiğini anlamalarını sağlıyor.

### *Düşük Kod ve Kodsuz Platformlar*

Karmaşık yazılım geliştirme süreçlerini ortadan kaldıran platformlar, daha fazla işletmenin makine öğrenmesi uygulamalarına kolayca erişebilmesini sağlıyor. Bu gelişme, girişimcilerin ve küçük işletmelerin yapay zekâ projelerini hızlı bir şekilde hayata geçirmesine olanak tanıyor ve AI tabanlı uygulamaların yaygınlaşmasını destekliyor. Bu alanda iki örnek verilebilir: Microsoft Power Apps, Google AppSheet. Microsoft düşük kodlu uygulama geliştirme platformu Power Apps ile, kullanıcıların minimum kodlama bilgisiyle uygulamalar oluşturmasına imkan tanıyor. Google’ın AppSheet platformu, kodlama bilgisi olmadan mobil ve web uygulamaları oluşturulmasını sağlayan bir kodsuz platformdur. Kullanıcılar, AI ve makine öğrenmesi araçlarını entegre ederek uygulamalarını daha akıllı hale getirebiliyorlar.

### *Sorumlu Makine Öğrenmesi ve Etik AI*

Algoritmaların adil ve tarafsız bir şekilde çalışması için yapılan etik geliştirmeler, özellikle ayrımcılığı ve önyargıları önlemeye yönelik büyük adımlar atılmasına yardımcı oluyor. Bu trend, özellikle hukuk, sağlık ve işe alım gibi kritik alanlarda daha adil fırsatlar yaratmayı hedefliyor. Toplumsal etkilerin değerlendirildiği bu alanda, şeffaflık ve eşitlik sağlanmak isteniyor. Bu alanda Microsoft ve Google şirketlerinin yaptıkları örnek verilebilir. Google AI Principles: Google, etik yapay zekâ kullanımını sağlamak için AI ilkelerini belirledi. Bu ilkeler doğrultusunda, şirket, AI sistemlerinin şeffaf, adil ve güvenli olmasını sağlamak için düzenlemeler getiriyor. Microsoft AI Ethics: Microsoft, AI etiği konusunda bir dizi girişim başlattı. Şirket, AI ve makine öğrenmesi uygulamalarında adalet, güvenlik ve şeffaflık ilkelerine dayalı çalışmalar yürütüyor. Özellikle "AI Fairness" raporları ile önyargıların önüne geçmeyi hedefliyor.

### *Kuantum Makine Öğrenmesi*

Kuantum bilgisayarlar, büyük veri setlerinin işlenmesi ve optimizasyon gibi alanlarda devrim yaratma potansiyeline sahip. Kuantum makine öğrenmesi, özellikle karmaşık simülasyonlar ve büyük ölçekli veri analizlerinde klasik bilgisayarların sınırlarını aşmayı vaat ediyor. Gelecekte sağlık, enerji ve finans gibi alanlarda önemli ilerlemelere yol açabilir. Bu alanda özellikle IBM ve Google güzel çalışmalar yürütüyor.

### Yapay Zekâ ve Nesnelerin İnterneti (IoT) Entegrasyonu

AI ve IoT'nin entegrasyonu, birçok sektörü dönüştürmeye devam ediyor. Akıllı şehirler, sağlık uygulamaları ve endüstriyel otomasyon gibi alanlarda, gerçek zamanlı veri analizi ile daha verimli ve otomatik sistemler ortaya çıkıyor. Örneğin, sağlık alanında IoT cihazları ile yapay zekâ sistemleri, hastaların sağlık durumlarını anlık olarak izleyebilir ve erken müdahale yapılmasını sağlayabilir. Bu alanda Siemens MindSphere ve GE Predix platformları dikkat çekiyor. MindSphere platformu, IoT cihazlarıyla entegre edilmiş yapay zekâ uygulamaları sunarak endüstriyel verimliliği artırmayı amaçlıyor. General Electric’in Predix platformu; gerçek zamanlı veri toplama ve analiz ile enerji, üretim ve sağlık sektörlerinde verimlilik sağlıyor.

Bu gelişmeler, sadece teknolojinin gücünü artırmakla kalmıyor, aynı zamanda toplumsal fayda yaratmayı hedefliyor. Yapay zekâ ve makine öğrenmesinin daha geniş kesimlere hitap etmesi, insanların yaşam kalitesini artırmayı ve toplumsal eşitsizlikleri azaltmayı vaat ediyor.

## MAKİNE ÖĞRENMESİNİN GELECEKTEKİ POTANSİYELİ

Makine öğrenmesinin gelecekteki potansiyeli oldukça geniş ve derinlemesine etkilere sahiptir. Sağlık sektöründe yapay zekâ destekli sistemler hastalıkların erken teşhisinde, genetik analizlerde ve kişiye özel tedavi yöntemlerinin geliştirilmesinde önemli rol oynayacaktır. Örneğin, kanser gibi karmaşık hastalıkların biyomarkerleri, büyük veri ve makine öğrenmesi algoritmaları ile daha hızlı ve doğru bir şekilde analiz edilebilecek. Ayrıca, yapay zekâ, ilaç keşfi süreçlerini hızlandırarak yeni tedavi yöntemlerinin geliştirilmesinde devrim yaratabilir.

Otonom araçlar, gelecekte ulaşımı daha güvenli ve verimli hale getirecek. Trafik yönetimi, hız sınırları, park yeri tespiti gibi alanlarda makine öğrenmesi sayesinde şehirler daha akıllı hale gelecek. Ayrıca, enerji tüketimi ve karbon salınımını optimize ederek çevreye duyarlı çözümler üretebilir. Akıllı şehirler, enerjinin daha verimli kullanılmasını sağlarken, insanların yaşam kalitesini artıracak.

Finans sektörü de makine öğrenmesinin sağladığı potansiyeli en fazla kullanacak alanlardan biri olacak. Algoritmalar, yatırım kararlarını daha hızlı ve doğru bir şekilde alabilmek için büyük veri analitiği kullanacak. Ayrıca, dolandırıcılık tespiti ve kredi değerlendirme süreçlerinde daha güvenilir ve objektif sonuçlar elde edilecek.

Eğitimde ise kişiye özel öğrenme yöntemleri ve adaptif öğrenme sistemleriyle, öğrencilerin bireysel ihtiyaçlarına göre daha verimli eğitim çözümleri sunulacak. Öğretmenlerin ve eğitimcilerin öğrencilerin gelişimini daha doğru bir şekilde izleyebilmesini sağlayacak, bu da eğitimde daha eşit ve verimli bir deneyim yaratacaktır.

Tarım sektörü de makine öğrenmesinden faydalanacak. Akıllı tarım uygulamaları, toprak analizi, hava durumu tahminleri ve bitki sağlığı izleme gibi konularda makine öğrenmesi ile çiftçilere daha verimli ve sürdürülebilir üretim yöntemleri sunulacak. Bu hem verimliliği artıracak hem de doğal kaynakların daha verimli kullanılmasını sağlayacak.

Ancak tüm bu gelişmelerin yanında etik, güvenlik ve gizlilik gibi sorunların çözülmesi büyük önem taşımaktadır. Makine öğrenmesi sistemlerinin daha yaygın hale gelmesiyle birlikte, algoritmalardaki önyargılar, gizlilik ihlalleri ve güvenlik açıkları gibi riskler de artacaktır. Bu yüzden, gelecekteki gelişmelerin başarılı ve sürdürülebilir olabilmesi için bu meselelerin dikkatlice ele alınması gerekecek. Makine öğrenmesinin geleceği, insan yaşamını daha verimli, güvenli, sürdürülebilir ve adil hale getirme konusunda büyük fırsatlar sunmakta, ancak bu potansiyel, doğru etik ve güvenlik önlemleri ile şekillendirilmeli.

# Kaynakça

(tarih yok). miro.medium.com: https://miro.medium.com/v2/resize:fit:1400/1\*thlFXrLECSKP3EnJgBWgvw.png adresinden alındı

(tarih yok). images.javatpoint.com: https://images.javatpoint.com/tutorial/machine-learning/images/k-nearest-neighbor-algorithm-for-machine-learning2.png adresinden alındı

Akdogan, A. (2020, Haziran 4). *Uygulamalı Evrişimsel Sinir Ağları (Convolutional Neural Network)*. https://medium.com/bili%C5%9Fim-hareketi/uygulamal%C4%B1-evri%C5%9Fimsel-sinir-a%C4%9Flar%C4%B1-convolutional-neural-network-7d643eaeb6a7 adresinden alındı

Alçınar, B. (2021, Kasım 28). *Makine Öğrenimi Modeli Nedir?* https://www.technopat.net/2021/11/28/makine-ogrenimi-modeli-nedir/ adresinden alındı

Arslan, E. (tarih yok). *Makine Öğrenmesi*. cahitcengizhan.com: https://cahitcengizhan.com/makine-ogrenmesi/ adresinden alındı

*Aşırı Uydurma Nedir?* (tarih yok). https://aws.amazon.com/tr/what-is/overfitting/ adresinden alındı

Aydın, G. (2021, Ocak 5). *Makine Öğreniminde Overfitting ve Underfitting*. https://www.linkedin.com/pulse/makine-%C3%B6%C4%9Freniminde-overfitting-ve-underfitting-gizem-ayd%C4%B1n/ adresinden alındı

Aydın, M. (2022, Eylül 4). *Veri Türleri*. https://biyoinformatik.net/veri-turleri adresinden alındı

Babalık, N. (2023, Ağustos 14). *Gözetimsiz (Denetimsiz) Öğrenme Yöntemleri*. https://miuul.com/blog/gozetimsiz-ogrenme-yontemleri adresinden alındı

Barak, B. (2021, Ekim 15). *Makine Öğrenmesi Nedir?* www.siberegitmen.com: https://www.siberegitmen.com/makine-ogrenmesi-nedir/ adresinden alındı

Candan, H. (2021, Aralık 26). *Adım Adım Makine Öğrenmesi Bölüm 3 : Denetimsiz Öğrenme Nedir?* https://medium.com/machine-learning-t%C3%BCrkiye/ad%C4%B1m-ad%C4%B1m-makine-%C3%B6%C4%9Frenmesi-b%C3%B6l%C3%BCm-3-denetimsiz-%C3%B6%C4%9Frenme-nedir-f890ada49a40 adresinden alındı

Cangüven, G. (2019, Eylül 20). *DENGESİZ VERİ KÜMELERİ İLE MAKİNE ÖĞRENMESİ*. https://medium.com/@g.canguven11/dengesi%CC%87z-veri%CC%87-k%C3%BCmeleri%CC%87-i%CC%87le-maki%CC%87ne-%C3%B6%C4%9Frenmesi%CC%87-63bbac5f6869 adresinden alındı

Chorev, S. (2024, Haziran 7). *How to Test Machine Learning Models*. https://www.deepchecks.com/how-to-test-machine-learning-models/ adresinden alındı

*Classification Algorithm.* (tarih yok). cdn.botpenguin.com: https://cdn.botpenguin.com/assets/website/Classification\_Algorithm\_f6b45f3f99.png adresinden alındı

*databasecamp.de.* (tarih yok). https://databasecamp.de/wp-content/uploads/naive-bayes-overview-1024x709.png adresinden alındı

DataLabTR. (2019, Nisan 16). *https://www.bing.com/images/search?view=detailV2&ccid=42tYLxw8&id=5B9B2E79E23B8DC2C09CAE69112643C7940FAF6E&thid=OIP.42tYLxw8S\_G02e7jexC5qQHaEo&mediaurl=https%3A%2F%2Fwww.metrixadigital.com.au%2Fwp-content%2Fuploads%2F2023%2F05%2FData-1-1024x640.png&cdnurl*. https://medium.com/@datalabtr/makine-%C3%B6%C4%9Frenmesi-i%C3%A7in-%C3%B6rnek-veri-k%C3%BCmeleri-65809555f2af adresinden alındı

*DataSet Iconic Photo.* (tarih yok). https://www.yandex.com.tr/gorsel/search?img\_url=https%3A%2F%2Fbanner2.cleanpng.com%2F20180624%2Fluz%2Fkisspng-data-set-diagram-chart-data-set-5b2f1f82750420.1125240015298149144793.jpg&lr=115704&pos=0&rpt=simage&text=Veri%20seti%20 adresinden alındı

*Decision Tree.* (tarih yok). miro.medium.com: https://miro.medium.com/v2/resize:fit:688/1\*bcLAJfWN2GpVQNTVOCrrvw.png adresinden alındı

*Denetimli ve Denetimsiz Öğrenme Arasındaki Fark Nedir?* (tarih yok). https://aws.amazon.com/tr/compare/the-difference-between-machine-learning-supervised-and-unsupervised/ adresinden alındı

*Denetimsiz Öğrenme*. (tarih yok). https://www.ultralytics.com/tr/glossary/unsupervised-learning adresinden alındı

*Derin Öğrenme nedir?* (tarih yok). https://www.oracle.com/tr/artificial-intelligence/machine-learning/what-is-deep-learning/ adresinden alındı

Donges, N. (tarih yok). *What Is Transfer Learning? Exploring the Popular Deep Learning Approach.* https://builtin.com/data-science/transfer-learning adresinden alındı

Ergin Tosunoğlu, R. Y. (2021). *Eğitimde Makine Öğrenmesi: AraĢtırmalardaki Güncel* . dergipark.org.tr: https://dergipark.org.tr/tr/download/article-file/1873550 adresinden alındı

Gökgöz, M. (2023, Nisan 5). *Özellik Mühendisliği (Feature Engineering)*. https://medium.com/@melekgokgoz/%C3%B6zellik-m%C3%BChendisli%C4%9Fi-feature-engineering-3c1596692c8e#:~:text=%C3%96zellik%20m%C3%BChendisli%C4%9Fi%2C%20ham%20verileri%20tahmine%20dayal%C4%B1%20modellere%20g%C3%B6re,da%20g%C3%B6r%C3%BCnmeyen%20veriler%20%C adresinden alındı

*Gözetimli Öğrenme*. (2023, Kasım 27). https://tr.wikipedia.org/wiki/G%C3%B6zetimli\_%C3%B6%C4%9Frenme adresinden alındı

Gürbüz, F. (2020, Mayıs 18). *Tekrarlayan Sinir Ağı -Recurrent Neural Networks (RNN)*. https://medium.com/@batincangurbuz/tekrarlayan-sinir-a%C4%9F%C4%B1-recurrent-neural-networks-rnn-17b517dd0b3e adresinden alındı

Hodgson, J. (2022, Kasım 9). *The 5 Stages of Machine Learning Validation*. https://towardsdatascience.com/the-5-stages-of-machine-learning-validation-162193f8e5db adresinden alındı

*How Does Machine Learning Work.* (tarih yok). images.spiceworks.com: https://images.spiceworks.com/wp-content/uploads/2022/04/04094749/5-6-e1715637473263.png adresinden alındı

*Human-level control through deep reinforcement learning*. (2015, Şubat 25). https://www.nature.com/articles/nature14236 adresinden alındı

İlkbahar, N. (2019, Ekim 25). *Data Preprocessing(Veri Ön İşleme)*. https://medium.com/@ilkbaharnaz/data-preprocessing-veri-%C3%B6n-i%CC%87%C5%9Fleme-85236484f913 adresinden alındı

Jaadi, Z. (tarih yok). *Principal Component Analysis (PCA): A Step-by-Step Explanation*. https://builtin.com/data-science/step-step-explanation-principal-component-analysis adresinden alındı

Jacinto, A. (24, Ocak 26). *10 Trends and Innovations in Machine Learning in 2024*. www.startechup.com: https://www.startechup.com/blog/10-trends-in-machine-learning-in-2024/ adresinden alındı

Jain, A. (2024, Mayıs 22). *20 Machine Learning Trends You Must Know in 2024*. digicrusader.com: https://digicrusader.com/machine-learning-trends/ adresinden alındı

Karabay, B. (2022, Haziran 7). *Makine Öğrenimi Algoritmaları*. https://berfinkarabay.medium.com/makine-ogrenimi-algoritmalari-75f378170669 adresinden alındı

Karakuş, D. C. (tarih yok). *Makine Öğrenmesi Algoritmaları* . ckk.com.tr: https://ckk.com.tr/Kitap/M%C3%BChendislik/ML%2001%20Makine%20%C3%96%C4%9Frenmesi%20Algoritmalar%C4%B1.pdf adresinden alındı

Lab, N. R. (2020, Ocak 8). *Öğrenme Aktarımı/Transfer Learning*. https://medium.com/novaresearchlab/%C3%B6%C4%9Frenme-aktar%C4%B1m%C4%B1-transfer-learning-c0b8126965c4 adresinden alındı

*Machine Learning Models*. (tarih yok). https://www.javatpoint.com/machine-learning-models adresinden alındı

*Machine Learning Models: What They Are and How to Build Them*. (2023, Kasım 29). https://www.coursera.org/articles/machine-learning-models adresinden alındı

*Machine Learning Services*. (tarih yok). www.oracle.com: https://www.oracle.com/tr/artificial-intelligence/machine-learning/what-is-machine-learning/ adresinden alındı

*Makine Öğrenimi*. (2024, Mayıs 29). tr.wikipedia.org: https://tr.wikipedia.org/wiki/Makine\_%C3%B6%C4%9Frenimi adresinden alındı

*Makine Öğrenimi Algoritmaları*. (2024, Eylül 3). www.geeksforgeeks.org: https://www.geeksforgeeks.org/machine-learning-algorithms/ adresinden alındı

*Makine Öğrenimi Nedir?* (tarih yok). aws.amazon.com: https://aws.amazon.com/tr/what-is/machine-learning/ adresinden alındı

*Makine Öğrenimi: Makine Öğreniminin Aşırı Uyumu*. (tarih yok). https://www.easiio.com/tr/easiio-machine-learning-overfitting/ adresinden alındı

*Makine Öğrenme Modelleri*. (2024, Ağustos 8). https://www.geeksforgeeks.org/machine-learning-models/ adresinden alındı

*Makine Öğrenmesi*. (tarih yok). www.netser.com.tr: https://www.netser.com.tr/tr/blog/makine-ogrenmesi adresinden alındı

*Makine Öğrenmesi*. (tarih yok). https://adm.atauni.edu.tr/pluginfile.php/4608/mod\_resource/content/1/Lojistik%20Bilgi%20Sistemleri%2012.%20Hafta%20Ders%20Notu.pdf adresinden alındı

*Makine öğrenmesi algoritmaları*. (tarih yok). azure.microsoft.com: https://azure.microsoft.com/tr-tr/resources/cloud-computing-dictionary/what-are-machine-learning-algorithms/#:~:text=Makine%20%C3%B6%C4%9Frenmesi%20algoritmalar%C4%B1%2C%20ki%C5%9Filerin%2C%20karma%C5%9F%C4%B1k%20veri%20k%C3%BCmelerini%20ke%C5%9Ffetmesi%2 adresinden alındı

*Makine Öğrenmesi Nedir?* (2024, Ekİm 1). blog.myunilab.ne: https://blog.myunilab.net/makine-ogrenmesi-nedir/#Makine\_Ogrenimi\_Nasil\_Calisir adresinden alındı

*Makine Öğrenmesi Veri Setleri*. (tarih yok). https://www.educba.com/machine-learning-datasets/ adresinden alındı

*Makine Öğrenmesi(Machine Learning) Nedir, Nasıl Çalışır?* (2022, Nisan 7). https://dqturkiye.com/blog/makine-ogrenmesimachine-learning-nedir-nasil-calisir/ adresinden alındı

Manika. (2024, Ekim 28). *How to Train a Machine Learning Model: The Complete Guide*. https://www.projectpro.io/article/training-a-machine-learning-model/936 adresinden alındı

Meşeci, E. (2021, Ağustos 9). *Veri Seti Nedir ve Nerede bulunur?* https://elifmeseci.medium.com/veri-seti-nedir-ve-nerede-bulunur-d825351b9f08 adresinden alındı

*Model Doğrulaması Nedir ve Neden Önemlidir?* (2024, Mayıs 25). https://www.geeksforgeeks.org/what-is-model-validation-and-why-is-it-important/ adresinden alındı

*Neural Network.* (tarih yok). www.investopedia.com: https://www.investopedia.com/thmb/5-hnhHpOzLM2GVXPlSstg8tJYLw=/1500x0/filters:no\_upscale():max\_bytes(150000):strip\_icc()/dotdash\_Final\_Neural\_Network\_Apr\_2020-01-5f4088dfda4c49d99a4d927c9a3a5ba0.jpg adresinden alındı

Noyan, M. (2019, Eylül 19). *Yeni Başlayanlar için Makine Öğrenmesi Algoritmaları*. https://merveenoyan.medium.com/yeni-ba%C5%9Flayanlar-i%C3%A7in-makine-%C3%B6%C4%9Frenmesi-algoritmalar%C4%B1-6b89b3a67750 adresinden alındı

Oleszak, M. (2024, Eylül 27). *Adversarial Machine Learning: Defense Strategies*. neptune.ai: https://neptune.ai/blog/adversarial-machine-learning-defense-strategies adresinden alındı

Öğündür, G. (2020, Haziran 9). *Overfitting (Aşırı Öğrenme), Underfitting (Eksik Öğrenme) ve Bias-Variance Çelişkisi*. https://medium.com/@gulcanogundur/overfitting-a%C5%9F%C4%B1r%C4%B1-%C3%B6%C4%9Frenme-underfitting-eksik-%C3%B6%C4%9Frenme-ve-bias-variance-%C3%A7eli%C5%9Fkisi-b92bef2f770d#:~:text=A%C5%9F%C4%B1r%C4%B1%20%C3%B6%C4%9Frenmenin%20aksine%2C%20bir%20model%20yet adresinden alındı

Özcan, T. (2024, Haziran 30). *Makine Öğrenmesinde Overfitting: Modelin Tuzağına Düşmek*. https://medium.com/@c.taha.ozcan/makine-%C3%B6%C4%9Frenmesinde-overfitting-modelin-tuza%C4%9F%C4%B1na-d%C3%BC%C5%9Fmek-c1f0e1910971#:~:text=Overfitting%2C%20makine%20%C3%B6%C4%9Frenmesinde%20bir%20modelin%20e%C4%9Fitim%20verilerine%20a%C5%9F%C4%B1r%C4%B1, adresinden alındı

*Özellik mühendisliği nedir?* (tarih yok). aws.amazon.com: https://aws.amazon.com/tr/what-is/feature-engineering/ adresinden alındı

Rabiakeser. (2023, Ekim 19). *Yapay Sinir Ağları: Makine Öğrenmenin Temel Taşı*. https://medium.com/@rabiakkeser/yapay-sinir-a%C4%9Flar%C4%B1-makine-%C3%B6%C4%9Frenmenin-temel-ta%C5%9F%C4%B1-5ad3be94b7e8 adresinden alındı

Raghavan, H. (2024, Temmuz 18). *What is a dataset in machine learning?* https://cloud2data.com/what-is-a-dataset-in-machine-learning/ adresinden alındı

Sakal, D. M. (2020, Haziran 20). *Makine Öğrenmesi Algoritmaları Kısa Açıklamaları*. muratsakal.com: https://muratsakal.com/?p=230 adresinden alındı

Salmons, J. (2023, Ağustos 17). *What do researchers need to know about using datasets?* https://researchmethodscommunity.sagepub.com/blog/what-do-researchers-need-to-know-in-order-to-access-and-use-datasets adresinden alındı

ŞİRİN, E. (2017, Ağustos 13). *Büyük Veri Ön-İşleme (Makale Notları)*. https://www.veribilimiokulu.com/buyuk-veri-on-isleme-makale-notlari/ adresinden alındı

Tafralı, S. (2023, Eylül 19). *Pekiştirmeli Öğrenmeye (Reinforcement Learning) Kısa Bir Giriş*. https://miuul.com/blog/pekistirmeli-ogrenmeye-k%C4%B1sa-bir-giris adresinden alındı

Tavasali, S. (2024, Ekim 28). *10 Types of Machine Learning Algorithms and Models*. www.simplilearn.com: https://www.simplilearn.com/10-algorithms-machine-learning-engineers-need-to-know-article adresinden alındı

Tech, C. (2019, Haziran 2). *What are the Various Types of Data Sets Used in Machine Learning?* https://www.cogitotech.com/blog/what-are-the-various-types-of-data-sets-used-in-machine-learning/ adresinden alındı

UZUN, T. (tarih yok). *Veri Analizi Aşamaları 09 – Özellik Mühendisliği*. https://turkeruzun.com.tr/veri-analizi-ozellik-muhendisligi/ adresinden alındı

*Veri madenciliği nedir?* (tarih yok). https://aws.amazon.com/tr/what-is/data-mining/#:~:text=Veri%20madencili%C4%9Fi%2C%20b%C3%BCy%C3%BCk%20veri%20k%C3%BCmelerini%20i%C5%9Flemek%20ve%20ke%C5%9Ffetmek,ve%20y%C3%B6ntemleriyle%2C%20verilerindeki%20gizli%20d%C3%BCzenleri%20ve%20ili%C5%9Fkileri%20 adresinden alındı

*Veri Nedir?* (tarih yok). https://www.bing.com/images/search?view=detailV2&ccid=42tYLxw8&id=5B9B2E79E23B8DC2C09CAE69112643C7940FAF6E&thid=OIP.42tYLxw8S\_G02e7jexC5qQHaEo&mediaurl=https%3A%2F%2Fwww.metrixadigital.com.au%2Fwp-content%2Fuploads%2F2023%2F05%2FData-1-1024x640.png&cdnurl adresinden alındı

*Veri Nedir? Veri Tabanı Nedir?* (2020, Ocak 3). https://www.iienstitu.com/blog/veri-nedir adresinden alındı

Weedmark, D. (2023, Şubat 9). *Machine learning model training: What it is and why it’s important*. https://domino.ai/blog/what-is-machine-learning-model-training adresinden alındı

Yıldırım, E. (2020, Mayıs 2). *Yapay Sinir Ağı(Artificial Neural Network) Nedir?* https://www.veribilimiokulu.com/yapay-sinir-agiartificial-neural-network-nedir/ adresinden alındı