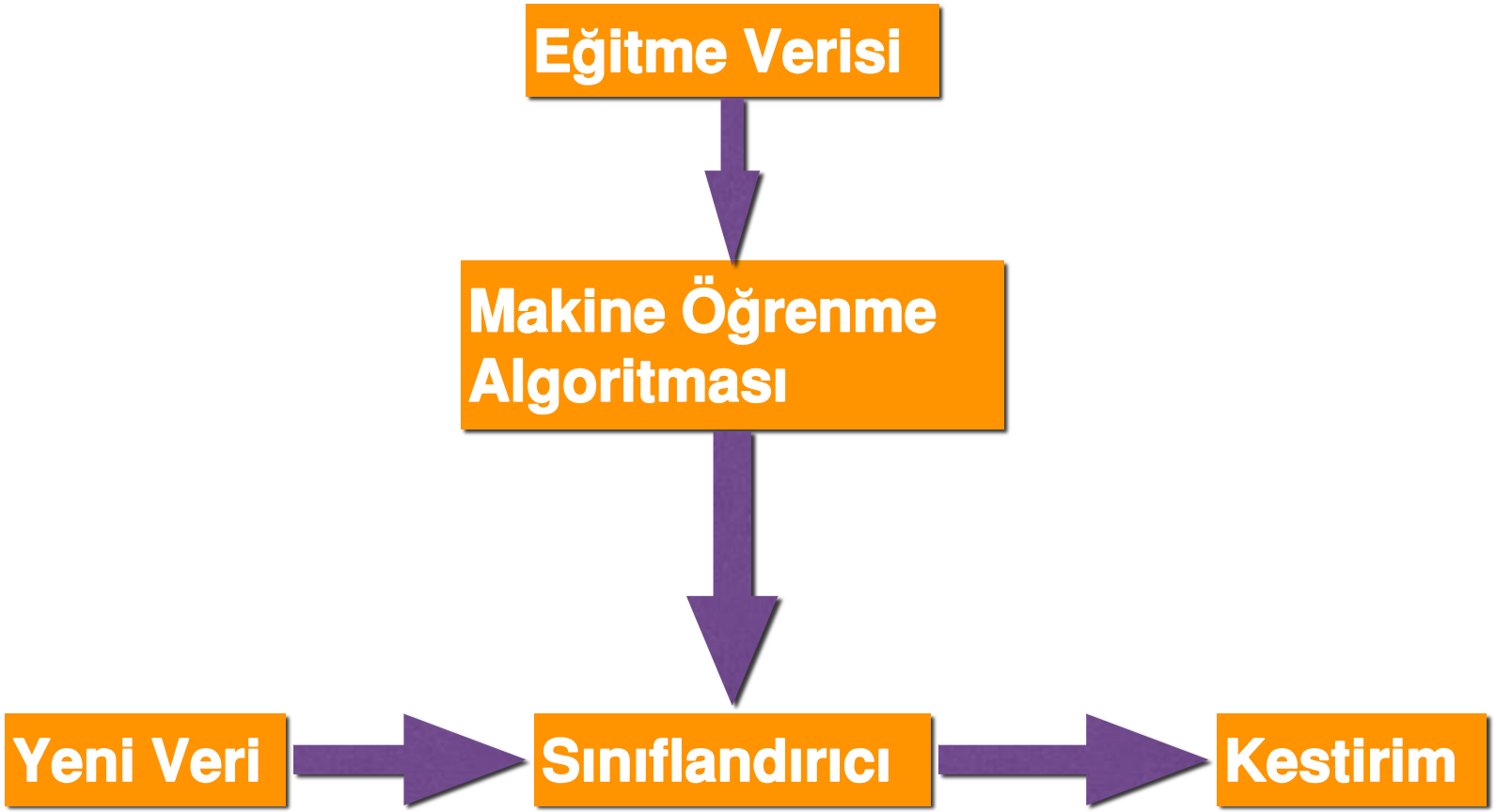
YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

Mekatronik Mühendisliği Uygulamalarında Yapay Zekâ

Ders 10- Makine Öğrenmesinde Sınıflandırma Algoritmaları

Prof. Dr. Erhan AKDOĞAN

Sınıflandırma:



Sınıflandırma Kavramı:

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

- Danışmanlı öğrenme olarak da adlandırılır.
- Etiketlenmiş veriler (eğitim verisi) üzerinden sistemin bir model oluşturması ve sonra etiketsiz (test kümesi) verilerinin sınıflarına göre etiketlenmesi işlemidir.

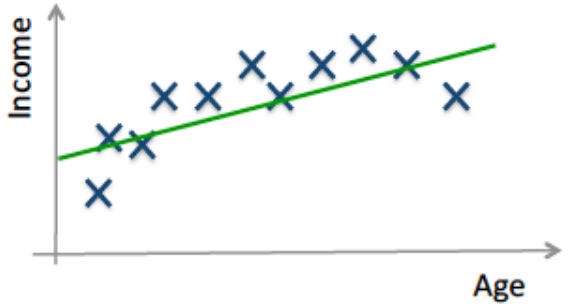
Sınıflandırma Türleri

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

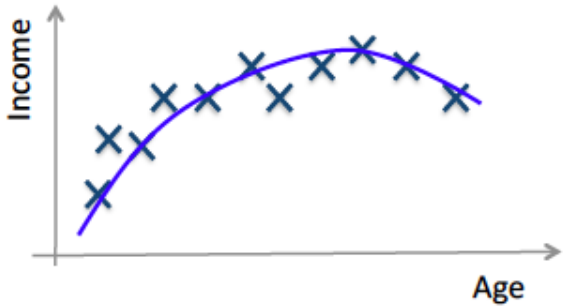
Sınıflandırma Türü	Avantajlar	Dezavantajlar	Uygulama Alanları
Lojistik Regresyon	Hızlı, kolay yorumlanabilir, doğrusal ilişkilerde etkili	Doğrusal olmayan ilişkilerde başarısız, aşırı öğrenmeye açık	Finansal tahminler, tıbbi teşhis, pazarlama analitiği
Destek Vektör Makineleri (SVM)	Yüksek boyutlu verilerle çalışabilir, doğrusal olmayan sınıflandırma yapabilir	Parametre ayarlaması zor, büyük veri setlerinde yavaş	Yüz tanıma, biyometrik kimlik doğrulama, gen analizi
Karar Ağaçları	Kolay anlaşılır, hızlı eğitim, veri ön işleme ihtiyacı az	Aşırı öğrenmeye açık, dengesiz veri setlerinde başarısız	Tıp, kredi risk değerlendirmesi, müşteri segmentasyonu
Rastgele Orman	Daha doğru sonuçlar, aşırı öğrenme riski düşük	Hesaplama maliyeti yüksek, model karmaşıklığı	Finans, biyoinformatik, görüntü işleme
Naive Bayes	Hızlı ve verimli, küçük veri setleriyle çalışabilir	Bağımsızlık varsayımı genelde gerçekçi değil	Spam filtreleme, metin sınıflandırma, medikal teşhis
K-En Yakın Komşu (KNN)	Parametrik olmayan, basit uygulama, öğrenmeye ihtiyaç duymaz	Büyük veri setlerinde performans düşebilir, k değerine bağlıdır	Öneri sistemleri, görsel tanıma, hareket tespiti
Yapay Sinir Ağları (ANN)	Karmaşık ilişkileri öğrenebilir, doğruluk oranı yüksek	Hesaplama maliyeti yüksek, fazla veri gerektirir	Ses tanıma, doğal dil işleme, görüntü işleme

Overfitting-Underfitting Kavramı:

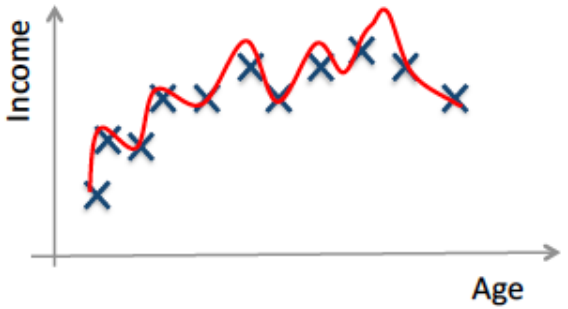
Yetersiz Uyum



High bias (underfitting)



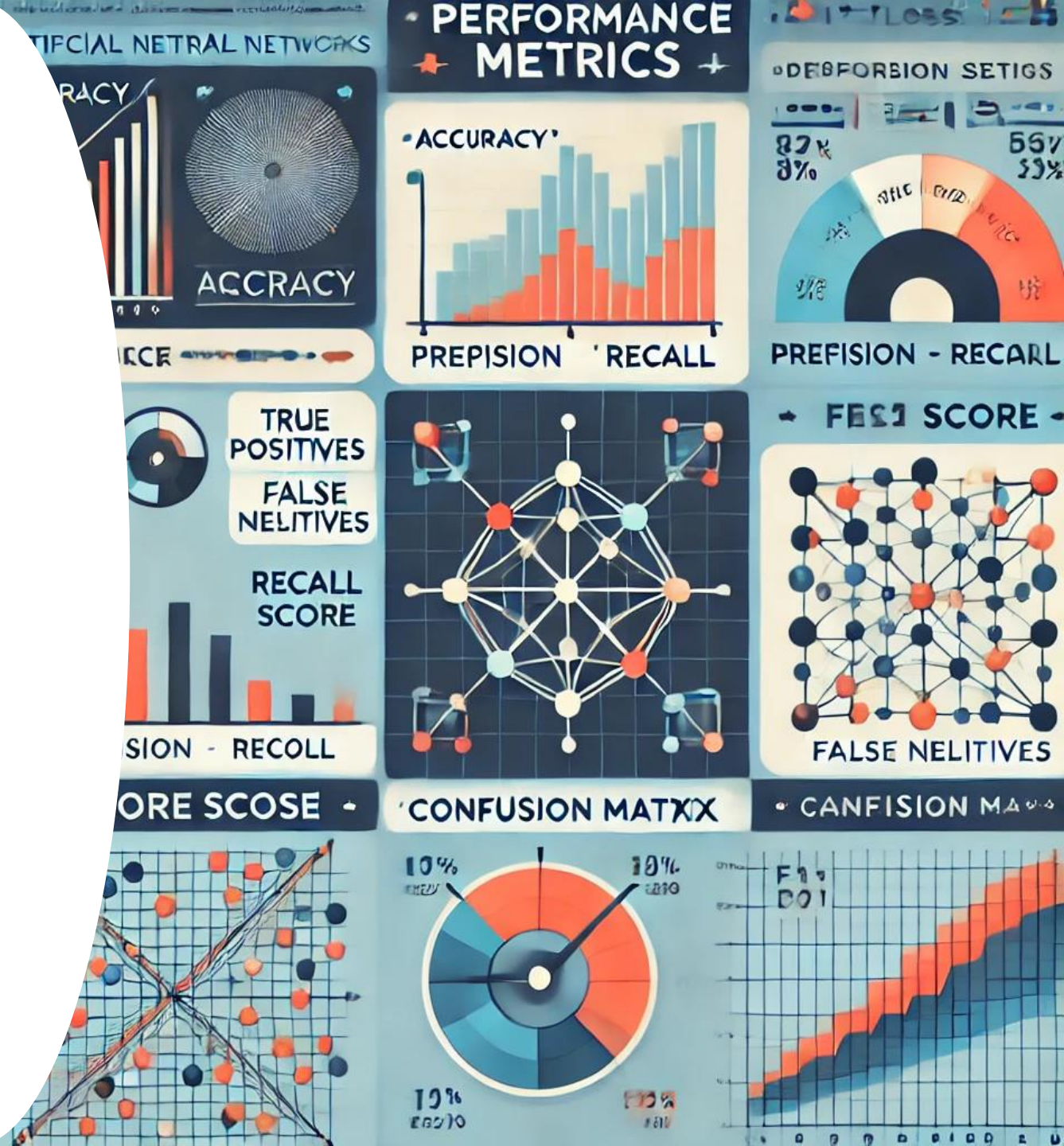
Just right!



Aşırı Uyum

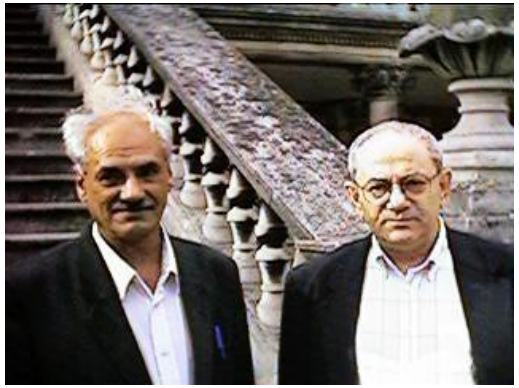
High variance (overfitting)

Destek Vektör Makineleri



1. Destek Vektör Makineleri (Support Vector Machine-SVM)

- Vladimir Vapnik ve Alexey Chervonenkis tarafından geliştirilmiştir (1963).
- Algoritmanın temel amacı, veriyi en iyi şekilde ayıran bir hiper düzlem bulmaktır.
- İstatiksel öğrenme teorisine dayanır.
- Yüksek boyutlu bir uzay haritalandırılır.
- İlk olarak optik karakter okumada kullanıldı.



Temel Prensibi

- SVM, veriyi, sınıflar arasında mümkün olan en geniş **marjinal mesafeyi** oluşturan bir hiper düzlem ile ayırır.
- "Destek vektörleri," bu hiper düzlemi belirlemekte kritik rol oynayan veri noktalarıdır. Hiper düzlemin pozisyonu, bu destek vektörlerine dayanır.

Avantajları:

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

- Yüksek boyutlu verilerde etkili çalışır.
- Doğrusal ve doğrusal olmayan verilere uygulanabilir.
- Doğrusal olmayan sınıflandırmalar için esneklik sunar.
- Marjinal mesafeyi maksimize ettiği için genelde iyi bir genelleştirme yapar.
- Dengesiz veri kümelerinde iyi sonuçlar verir
- Karmaşık karar sınırlarını modeller.
- Çok sayıda bağımsız değişken ile çalışabilir.
- Aşırı uyum sorunu azdır (göreceli olarak).

Özellikleri

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

1.Doğrusal ve Doğrusal Olmayan Sınıflandırma:

1. Verilerin **doğrusal olarak ayrılabildiği** durumlarda, SVM **doğrusal** bir hiper düzlem kullanır.
2. Verilerin **doğrusal olarak ayrılamadığı** durumlarda, SVM, **çekirdek (kernel) fonksiyonları** kullanarak veriyi yüksek boyutlu bir uzaya dönüştürür ve bu uzayda ayrım yapar.

2.Çekirdek Fonksiyonları:

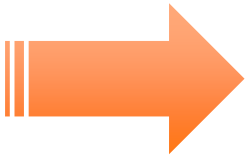
1. **Doğrusal Çekirdek**: Doğrusal ayrılabilir veriler için.
2. **Polinomal Çekirdek**: Daha karmaşık ayrımlar için.
3. **Radyal Taban Fonksiyonu (RBF)**: Genellikle doğrusal olmayan durumlarda kullanılır.
4. **Sigmoid Çekirdek**: Bazı durumlarda kullanışlıdır.

3.Hiper Parametreler:

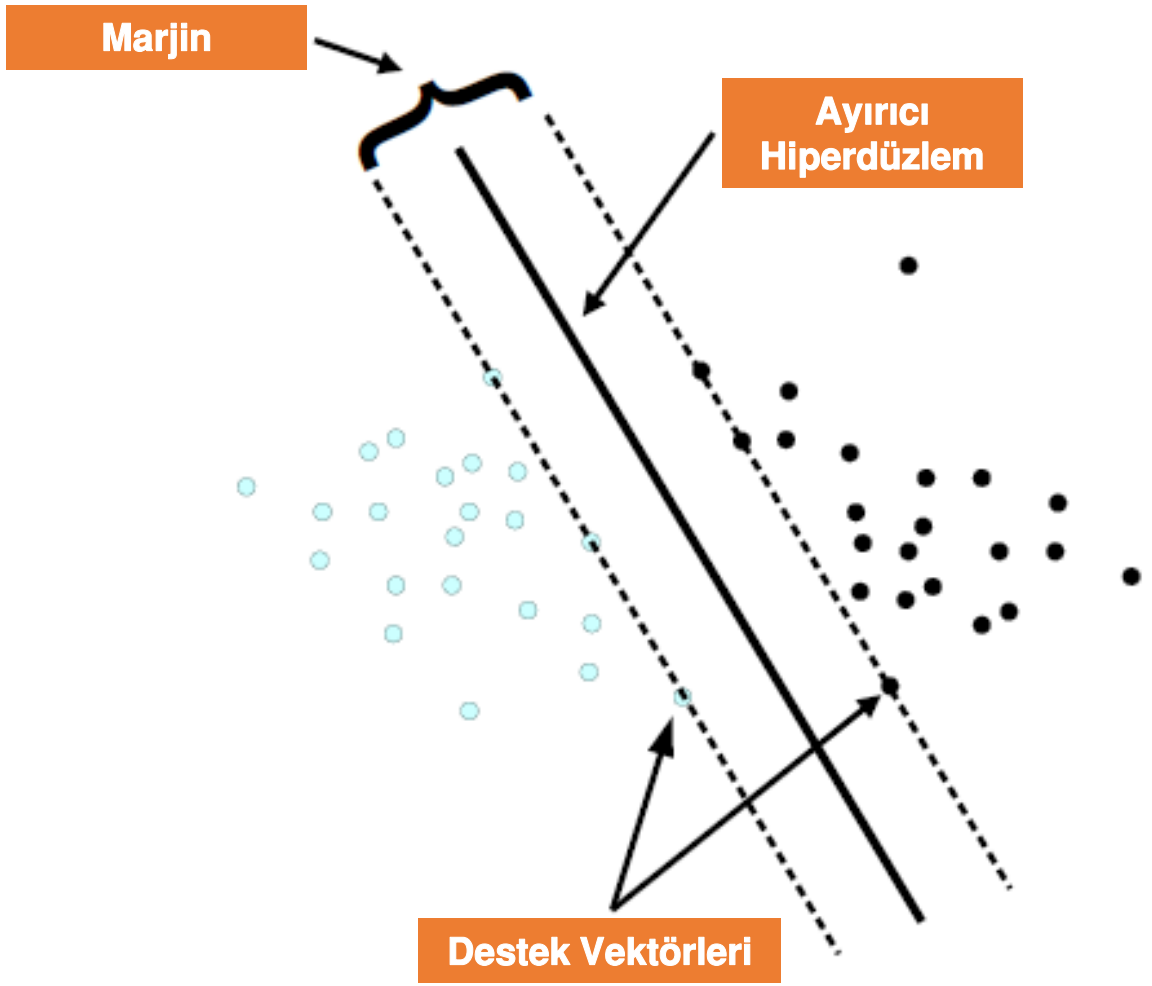
1. **C**: Hata toleransı ile marjinal mesafe arasındaki dengeyi kontrol eder.
2. **Gamma (γ)**: Çekirdek fonksiyonlarının etkisini kontrol eder (özellikle RBF ve polinomal çekirdeklerde).

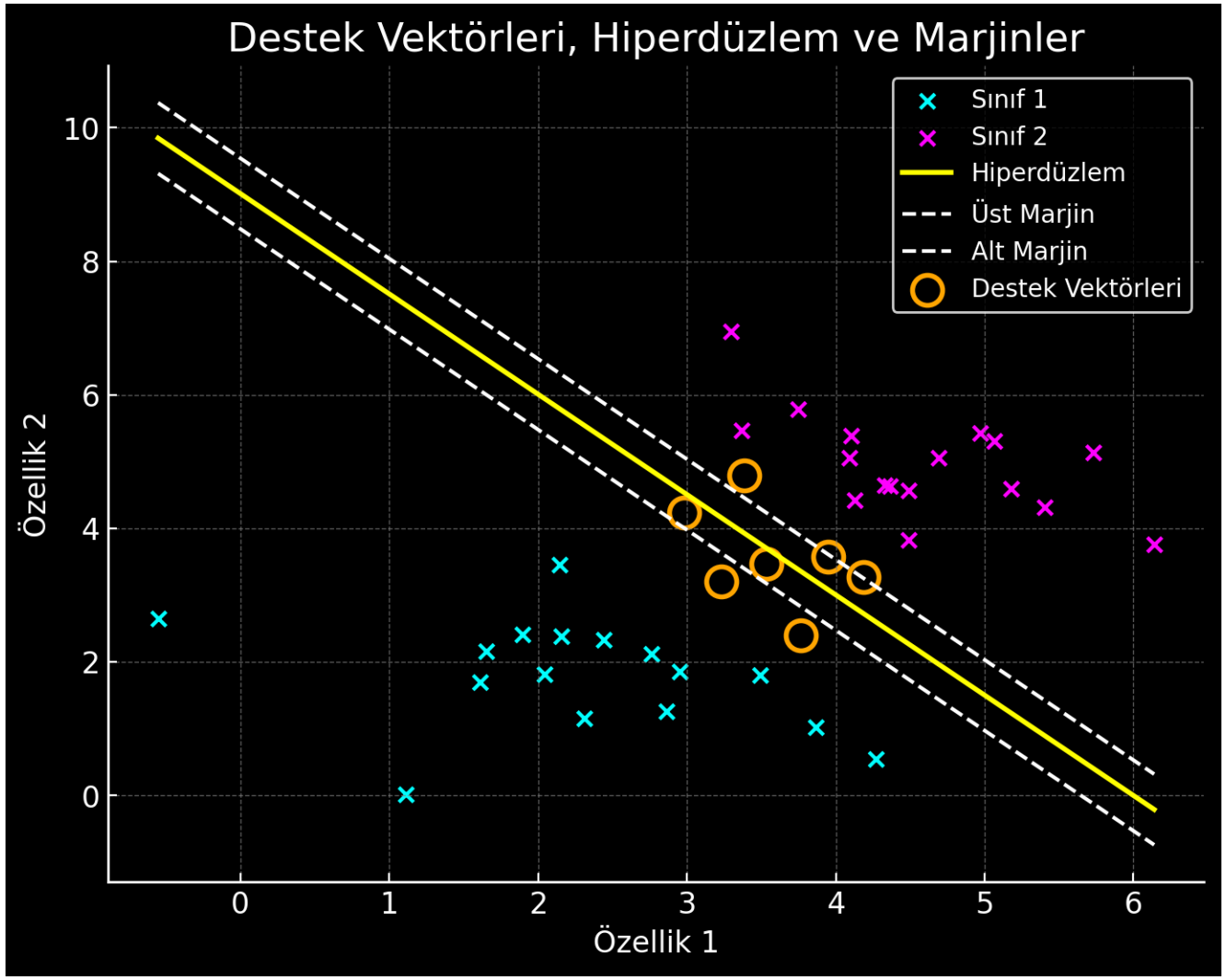
Amaç:

- Sınıf sayısına göre sınıf üyeleri arasında bu sınıfları birbirinden ayıran sonsuz sayıdaki doğru içinden marjini en yüksek doğruyu seçerek sınıflandırmayı gerçekleştirmek.
- Oluşan marjin doğrusu sınıf üyelerinin seçilen doğruya en yakın olan üyelerine paralel olmalıdır. Çizilen doğrulara **hiperdüzlem** denir.



Hiperdüzlem (hyperplane)





-Mavi ve kırmızı noktalar: İki farklı sınıfa ait veri noktalarını temsil eder.

-Yeşil çizgi: İki sınıfı en iyi şekilde ayıran **hiperdüzlemi** gösterir.

-Daire içindeki noktalar: SVM algoritmasının hiperdüzlemi belirlemek için kullandığı **destek vektörleridir**.

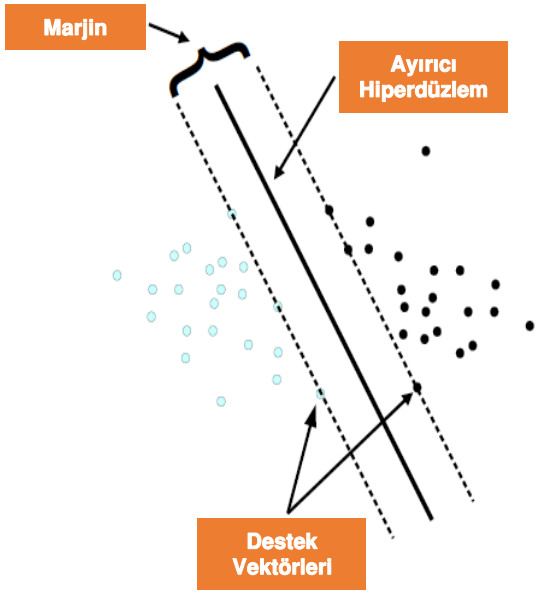
Hiperdüzlem, bu destek vektörlerine dayanarak optimize edilmiştir ve iki sınıf arasındaki marjı maksimize etme amacı taşır.

Destek Vektörleri

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

Destek vektörleri, **veri setindeki kritik veriler**dir. Bu noktalar, farklı sınıflar arasındaki ayırım sınırını (hiperdüzlemi) tanımlamak ve optimize etmek için kullanılır.

- Destek vektörleri, ayırıcı hiperdüzleme en yakın olan verilerden oluşur.
- Ayırıcı sınırın konumunu etkileyen yalnızca bu noktalar olduğu için, SVM'nin öğrenme sürecinde gereksiz verilerle ilgilenmez; bu, algoritmanın verimli çalışmasını sağlar.



Örnek:

İki sınıf arasında bir sınır çizmeye çalıştığınızı düşünün. Bu sınırın tam olarak nerede olacağını belirleyen, her sınıfa en yakın olan noktalar (destek vektörleri) olacaktır.

Hiperdüzlem

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

Hiperdüzlem, SVM'nin iki sınıfı ayırmak için kullandığı **ayırıcı sınırdır**.

- İki sınıf arasındaki mümkün olan **en geniş marjı** (boşluğu) sağlayan düzlemdir.
- Verilerin boyutuna bağlı olarak, bir düz çizgi (2 boyutta), bir düzlem (3 boyutta) veya daha yüksek boyutlarda bir hiperdüzlem olarak adlandırılır.

Özellikler:

- Hiperdüzlem, verileri olabildiğince doğru şekilde sınıflandırmayı amaçlar.
- Destek vektörlerinden gelen bilgilere dayanarak optimize edilir.

Destek Vektörleri ve Hiperdüzlem Arasındaki Fark

- **Destek Vektörleri:** Veriler arasındaki hiperdüzlemi belirleyen kritik noktalardır. Hiperdüzlem bu noktalara dayanarak optimize edilir.
- **Hiperdüzlem:** Destek vektörlerinden elde edilen bilgilere göre farklı sınıfları ayırmak için kullanılan düzlemdir.

Birlikte Çalışmaları:

Destek vektörleri, hiperdüzlemin doğru şekilde yerleşmesi için algoritmaya rehberlik eder. SVM'nin başarısı, bu hiperdüzlemin sınıfları olabildiğince iyi ayırmasına bağlıdır.

Bire karşı bir(BKB) ve Bire karşı diğer(BKD)

- Sınıflandırma “iki sınıf” temellidir.
- Daha çok sınıf var ise ikili sınıflamalar birleştirilir.
- Bunun için farklı yöntemler mevcuttur.
  - **Bire karşı bir(BKB)**
  - **Bire karşı diğer(BKD)**

  bilinen yöntemler arasındadır.

Bire karşı bir(BKB) ve Bire karşı diğer(BKD)

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

İkili Sınıflama (Binary Classification):

- İkili sınıflama, yalnızca iki sınıf arasında bir ayrım yapmayı içerir (örneğin, evet/hayır, hasta/sağlıklı gibi).

Çoklu Sınıflandırma (Multi-class Classification):

- Eğer sınıflandırılacak öğe iki sınıftan fazlaysa (örneğin, kırmızı/mavi/yeşil gibi), ikili sınıflama yaklaşımlarını birleştirerek çoklu sınıflandırma yapılabilir.

Bire karşı bir(BKB) ve Bire karşı diğer(BKD)

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

Bire Karşı Bir (BKB):

- Bu yöntemde, her bir sınıf diğer sınıflarla tek tek karşılaştırılır. Örneğin, sınıflar A, B, C ise:
  - A-B, A-C, ve B-C ikili sınıflamaları yapılır.
- Toplamda $n \times (n-1)/2$ adet model eğitilir (n sınıf sayısıdır).

Bire Karşı Diğer (BKD):

- Her bir sınıf, diğer tüm sınıflara karşı bir sınıflandırma modeli oluşturur. Örneğin:
  - A, diğerlerine karşı
  - B, diğerlerine karşı
  - C, diğerlerine karşı
- Bu yöntem, n adet model gerektirir (n sınıf sayısıdır).

Doğrusal olarak ayrılamayan veri kümelerinde sınıflandırma

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

- Doğrusal olmayan verilerle çalışma yeteneği. SVM, verilerin doğrusal olarak ayrılamadığı durumlarda, **çekirdek (kernel) fonksiyonları** kullanarak verileri yüksek boyutlu bir uzaya dönüştürür. Bu süreç, karmaşık veriler arasında ayrım yapmayı mümkün kılar.

Çekirdek (Kernel) Fonksiyonu Nedir?

Bir çekirdek fonksiyonu, iki veri noktasını (örneğin, x_1 ve x_2) yüksek boyutlu bir uzaya projeksiyon yapmadan, bu uzaydaki benzerliklerini hesaplar. Böylece, veri yüksek boyutlara taşınmış gibi işlem yapılabilir, ancak hesaplama maliyeti düşük kalır.

Çekirdek fonksiyonu:

$$K(x_1, x_2) = \phi(x_1) \cdot \phi(x_2)$$

Burada $\phi(x)$ veri noktalarını yüksek boyutlu bir uzaya dönüştüren bir fonksiyondur.

Sık Kullanılan Çekirdek (Kernel) Fonksiyonları

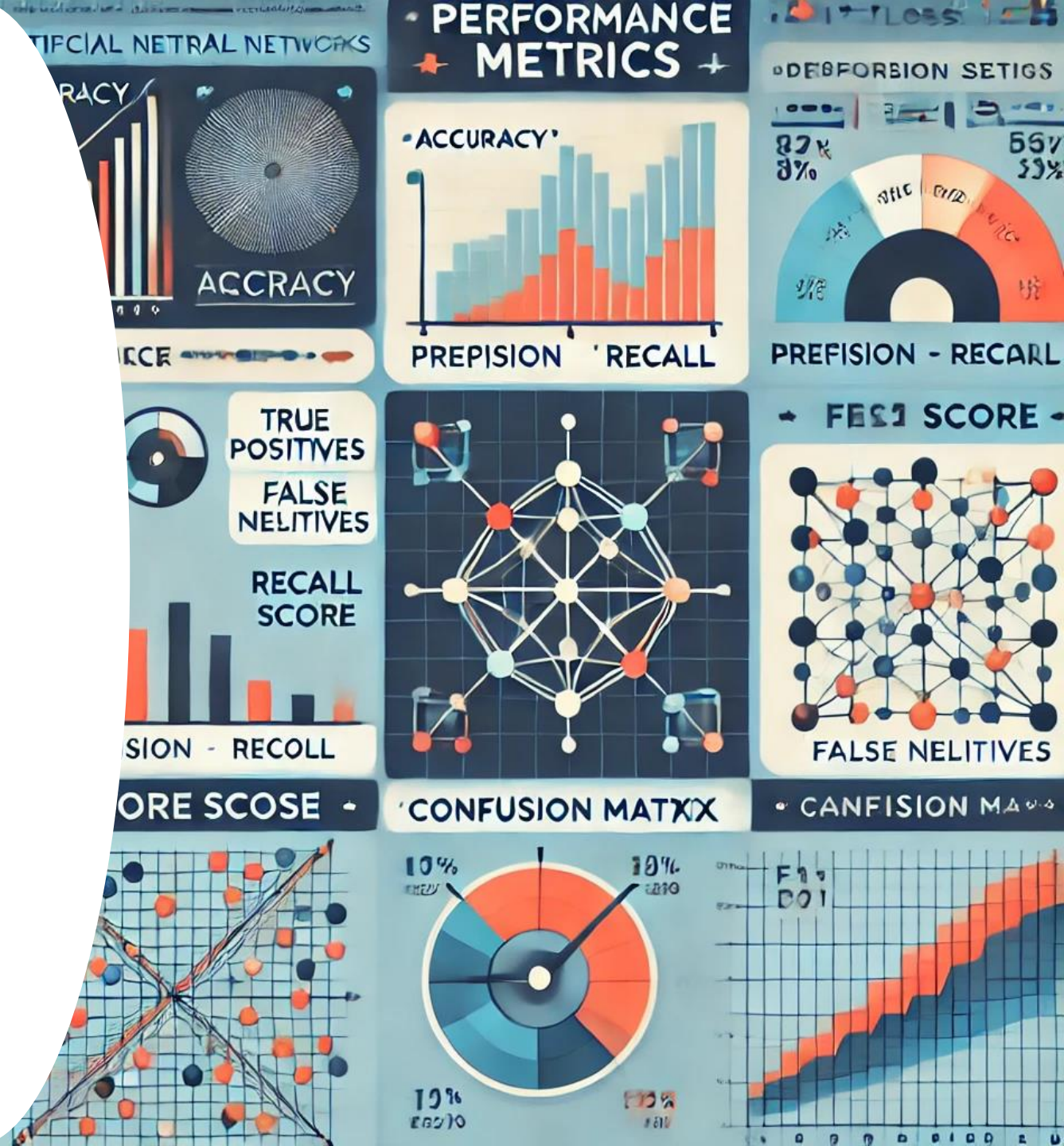
YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

1. **Doğrusal Çekirdek** ($K(x_1, x_2) = x_1 \cdot x_2$)
   - Veriler doğrusal olarak ayrılabiliyorsa kullanılır.
2. **Polinomal Çekirdek** ($K(x_1, x_2) = (x_1 \cdot x_2 + c)^d$)
   - Daha karmaşık sınıflandırmalar için uygundur.
   - d, polinom derecesini belirtir.
3. **Radyal Taban Fonksiyonu (RBF)** ($K(x_1, x_2) = \exp(-\gamma \|x_1 - x_2\|^2)$)
   - En yaygın kullanılan çekirdeklerden biridir.
   - Veri noktalarının uzaklığına bağlı olarak benzerlik hesaplar.
4. **Sigmoid Çekirdek** ($K(x_1, x_2) = \tanh(\alpha x_1 \cdot x_2 + c)$)
   - Sinir ağlarındaki aktivasyon fonksiyonlarına benzer.

Örnek : Bir dairesel sınıf ayrımı yapılması gereken durumda (örneğin, iç içe geçmiş iki sınıf), SVM doğrusal bir hiper düzlem bulamaz.

Ancak, RBF gibi bir çekirdek fonksiyonu kullanıldığında, veriler yüksek boyutlu bir uzaya yansıtılarak doğrusal hale getirilebilir ve SVM bu yeni uzayda sınıfları ayırabilir.

YSA’larda Sınıflandırma

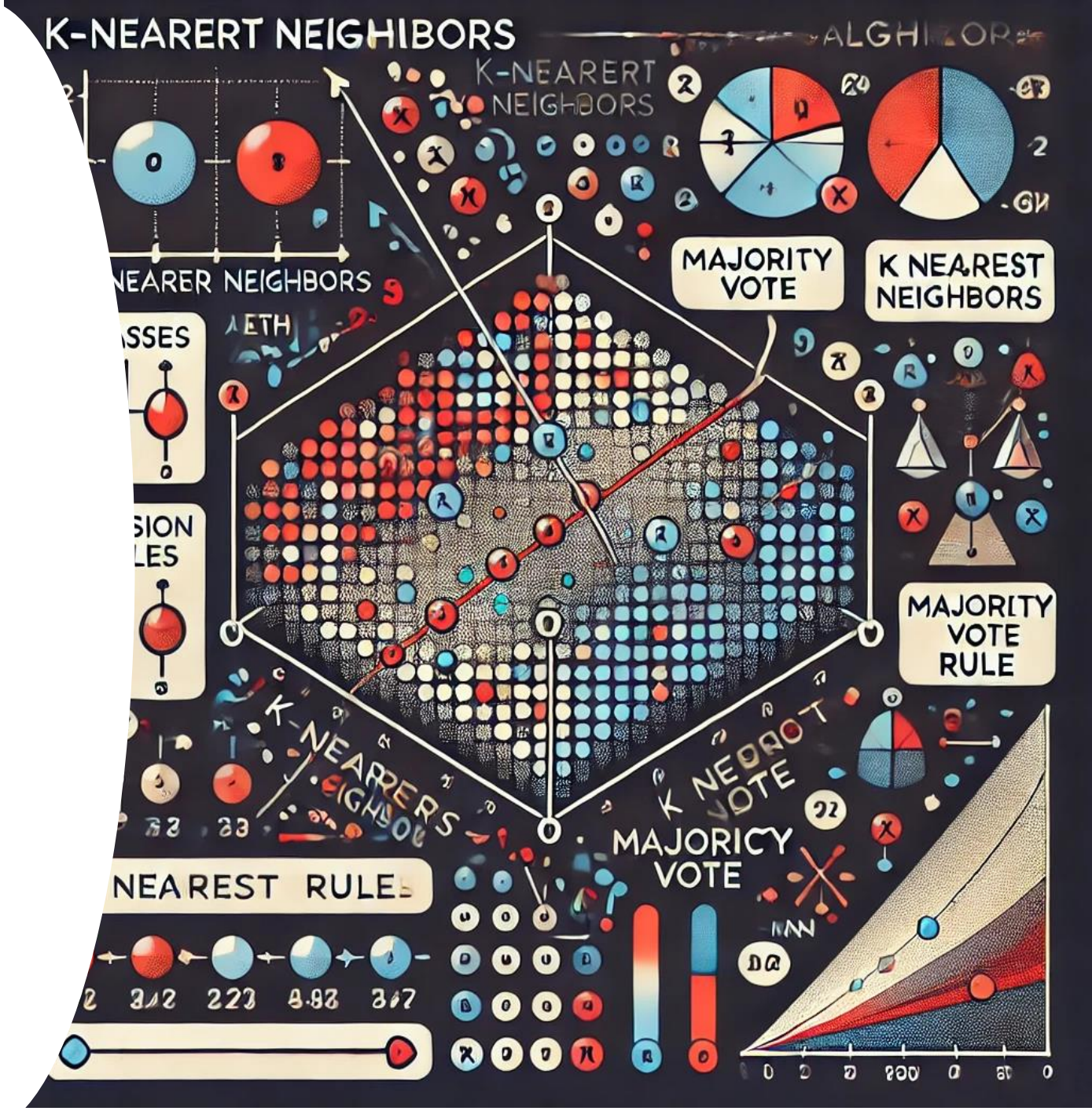


2. Yapay Sinir Ağlarında Sınıflandırma

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

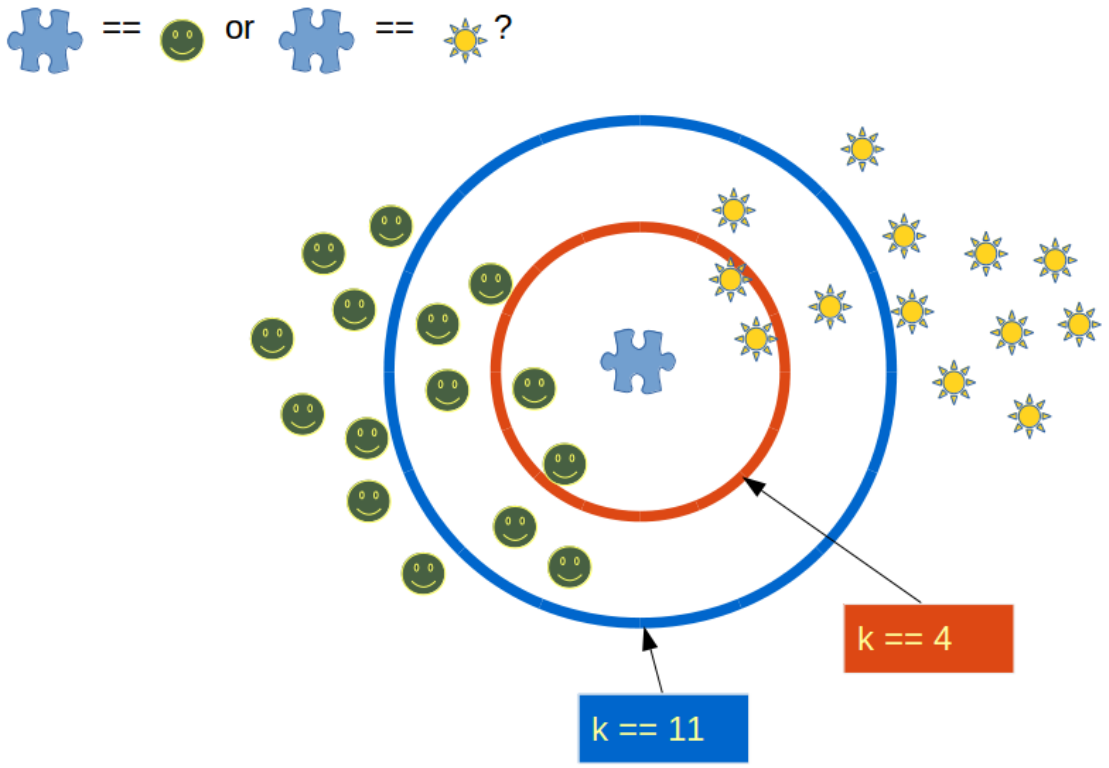
- Perceptron
- Çok Katmanlı Algılayıcı (Multi-Layer Perceptron - MLP)
- Radial Basis Function (RBF) Ağları
- Konvolüsyonel Sinir Ağları (Convolutional Neural Networks - CNN)
- Yinelemeli Sinir Ağları (Recurrent Neural Networks - RNN)
- Uzun Kısa Süreli Bellek (Long Short-Term Memory - LSTM)
- Gated Recurrent Unit (GRU)
- Otomatik Kodlayıcılar (Autoencoders)
- Self-Organizing Maps (SOM)

k-En Yakın Komşu



3. k-En Yakın Komşu (K-Nearest Neighbour)

- k-En Yakın Komşu (KNN), hem sınıflandırma hem de regresyon problemleri için kullanılan basit ve etkili bir gözetimli öğrenme algoritmasıdır.
- Seçilen bir özelliğin kendisine en yakın olan özellikle arasındaki yakınlığı kullanarak sınıflandırma gerçekleştirilir.
- ***k*** parametresi komşu sayısını temsil etmektedir.
- Temel prensibi, bir veri noktasını en yakın ***k*** komşusuna dayanarak sınıflandırmak veya tahmin etmektir.



Kullanım Alanları:

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

- Görüntü tanıma
- Metin sınıflandırma
- Tıp
- Anomali tespiti
- Öneri sistemleri
- Sahtecilik tespiti
- Hedefli pazarlama
- Zaman serisi analizi
- Genomik ve biyoinformatik
- Ses tanıma

Avantajları / Dezavantajları:

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

Avantajlar:

- Uygulanması kolaydır.
- Gürültülü verilerde etkili çalışabilir.

Dezavantajlar:

- Eğitim setinin büyüklüğü fazla ise daha etkin sonuçlar üretir.
- Algoritmanın başlangıcında ***k*** parametresine ihtiyaç duyar.
- En iyi sonucun elde edilebilmesi için hangi uzaklık ölçüsünün seçileceği belirgin değildir.
- Hesaplama maliyeti yüksektir.

Çalışma Metodu:

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

1. **Eğitim Süreci**:
   - KNN, eğitim sırasında herhangi bir model oluşturmaz. Tüm veri noktalarını bellekte saklar ve yeni bir veri geldiğinde karar verir. Bu nedenle, **model-free (modelsiz)** ve **lazy learning (tembel öğrenme)** yöntemlerinden biridir.

2. **Tahmin Süreci**:

   Yeni bir veri noktası için:
   - Veri noktası ile mevcut tüm eğitim verileri arasındaki mesafeyi hesaplar.
   - En yakın ***k*** komşuyu belirler.
   - Sınıflandırma problemlerinde, komşuların sınıflarına göre çoğunluğa dayalı bir karar verir.
   - Regresyon problemlerinde, komşuların ortalama değerini döndürür.

1. **Uzaklık Hesaplama**:

   Verinin doğasına göre uygun uzaklık ölçütü seçilir.

   En sık kullanılan uzaklık ölçümleri:

   - **Öklid Mesafesi**
   - **Manhattan Mesafesi**
   - **Minkowski Mesafesi** (Genelleştirilmiş form)

2. **En Yakın k Komşuyu Belirleme:**
   - Tüm veri noktalarıyla mesafeler hesaplanır ve sıralanır.
   - En yakın ***k*** veri noktası seçilir.

3. **Tahmin Yapma**:
   - **Sınıflandırma:** Komşuların çoğunluk oyu alınır.

k Değeri:

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

- **Çok küçük bir *k* :** Model gürültüye duyarlı olur (aşırı öğrenme).
- **Çok büyük bir *k* :** Model genelleştirme yeteneği düşer (az öğrenme).

3.1 Öklid Mesafesi:

- En yaygın kullanılan teknik.
- İki nokta arasındaki doğrusal mesafe olarak açıklanabilir.

$\{x_1, x_2, \ldots x_n\}$

$\{y_1, y_2, \ldots y_n\}$

→ **Herhangi iki nokta**

Öklid Mesafesi →

$$d = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}$$

→ Örnek

3.2. Manhattan Mesafesi:

- N boyutlu iki nokta arasındaki farkın mutlak değerlerinin toplamını ifade eder.

$\{x_1, x_2, \ldots x_n\}$

$\{y_1, y_2, \ldots y_n\}$

→ Herhangi iki nokta

Manhattan Mesafesi →

$$d = \sum_{i=1}^{k} | (x_i - y_i) |$$

3.2. Manhattan Mesafesi:

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

Manhattan mesafesi (veya L1 mesafesi), iki nokta arasındaki mesafeyi, yalnızca dikey ve yatay doğrultularda hareket ederek ölçen bir metrik türüdür.

Mekatronik uygulamalarında, genellikle robotik veya mobil sistemlerde rota planlama, engel algılama ve optimizasyon problemlerinde kullanılabilir.

Mekatronik Uygulamalar:

- **Depo Yönetimi:** Depo içindeki robotlar ürünleri toplamak için bu mesafeyi minimize ederek daha verimli rota planlayabilir.
- **Engel Algılama ve Rota Planlama:** Robotun Manhattan mesafesi kullanılarak engelleri dolaşıp en kısa yolu bulması sağlanabilir.
- **Robot Kolu Kontrolü:** Robotik kolların bir hedef noktaya ulaşırken dikey ve yatay düzlemlerde hareketini optimize etmek için kullanılabilir.

Bu tür bir hesaplama, özellikle basit ve hızlı algoritmalara ihtiyaç duyulan düşük işlem gücüne sahip sistemlerde faydalıdır.

3.3. Minkowski Mesafesi:

Mekatronik sistemlerde kullanılan yapay zeka algoritmalarında (örneğin, sınıflandırma veya kümeleme) Minkowski mesafesi, veriler arasındaki benzerlikleri ölçmek için bir metrik olarak kullanılır. Örneğin:

- Robotik bir sistemde öğrenme algoritmasıyla farklı yüzey türlerini tanımlamak.
- Sensör verilerini gruplandırarak anomali tespiti yapmak.

Minkowski Mesafesinin Avantajları

- **Esneklik**: *p* parametresini değiştirerek farklı türde mesafeleri modelleyebilir.
- **Çok Boyutluluk**: Üç boyutlu konumların veya sensör verilerinin karşılaştırılmasında etkili.
- **Kapsamlı Uygulama Alanı**: Robotik, yapay zeka, hata tespiti, sensör füzyonu gibi birçok alanda kullanılabilir.

3.3. Minkowski Mesafesi:

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

- Öklid ve Manhattan Mesafelerinin genelleştirilmiş halidir.

$\{x_1, x_2, \ldots x_n\}$

$\{y_1, y_2, \ldots y_n\}$

→ Herhangi iki nokta

Minkowski Mesafesi →

$$d = \left(\sum_{i=1}^{k} (| (x_i - y_i) |)^p \right)^{\frac{1}{p}}$$

$p = 1$ Manhattan
$p = 2$ Öklid

→ Örnek

Öklid, Manhattan ve Minkowski Karşılaştırma

Özellikler	Minkowski	Öklid	Manhattan
Mesafe Hesaplama Yöntemi	Genel bir mesafe ölçütü, p parametresi ile özelleştirilebilir.	$p = 2$ durumunda Minkowski mesafesinin özel hali.	$p = 1$ durumunda Minkowski mesafesinin özel hali.
Uygunluk	Hem Manhattan hem de Öklid mesafesi için uygun.	En kısa fiziksel mesafeyi hesaplamak için uygun.	Izgara tipi (grid) verilerde daha iyi performans gösterir.
Kullanım Alanları	Esnek kullanım ($p=1$: Manhattan, $p=2$: Öklid).	Coğrafi konumlar, robotik rotalar, 3B nesneler.	Şehir planlaması, ağ analizi, robot rotaları.
Avantajları	Her türlü veri dağılımına uyarlanabilir.	Basit ve sezgisel.	Aykırı değerlere karşı daha dayanıklı.
Dezavantajları	p parametresinin belirlenmesi bazen zor olabilir.	Aykırı değerlerden çok etkilenir.	Doğrudan çizgisel mesafeleri ölçemez.

Test örneğinin hangi sınıfa ait olduğuna karar verme

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

1. Komşu Veri Noktalarının Seçilmesi

- k parametresi, kaç tane en yakın komşunun dikkate alınacağını belirtir.
- Test örneği ile tüm eğitim veri noktaları arasındaki mesafe hesaplanır. Genellikle **Öklid mesafesi** kullanılır.
- Mesafeler küçükten büyüğe sıralanır ve en küçük k adet veri noktası seçilir.

2. Komşu Sınıfların Belirlenmesi

- Seçilen k adet veri noktasının sınıfları kontrol edilir.
- Her komşunun sınıfı, karar vermede eşit öneme sahiptir (varsayılan KNN).

Test örneğinin hangi sınıfa ait olduğuna karar verme

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

3. Çoğunluk Oyu Yöntemi

- Komşu sınıflar arasında hangi sınıfın daha fazla temsil edildiğine bakılır.
- Örneğin, $\boldsymbol{k = 5}$ ise ve seçilen komşuların sınıf dağılımı şu şekildeyse:

  $Sınıf\ 1 \rightarrow\ 3\ komşu$

  $Sınıf\ 2 \rightarrow\ 2\ komşu$

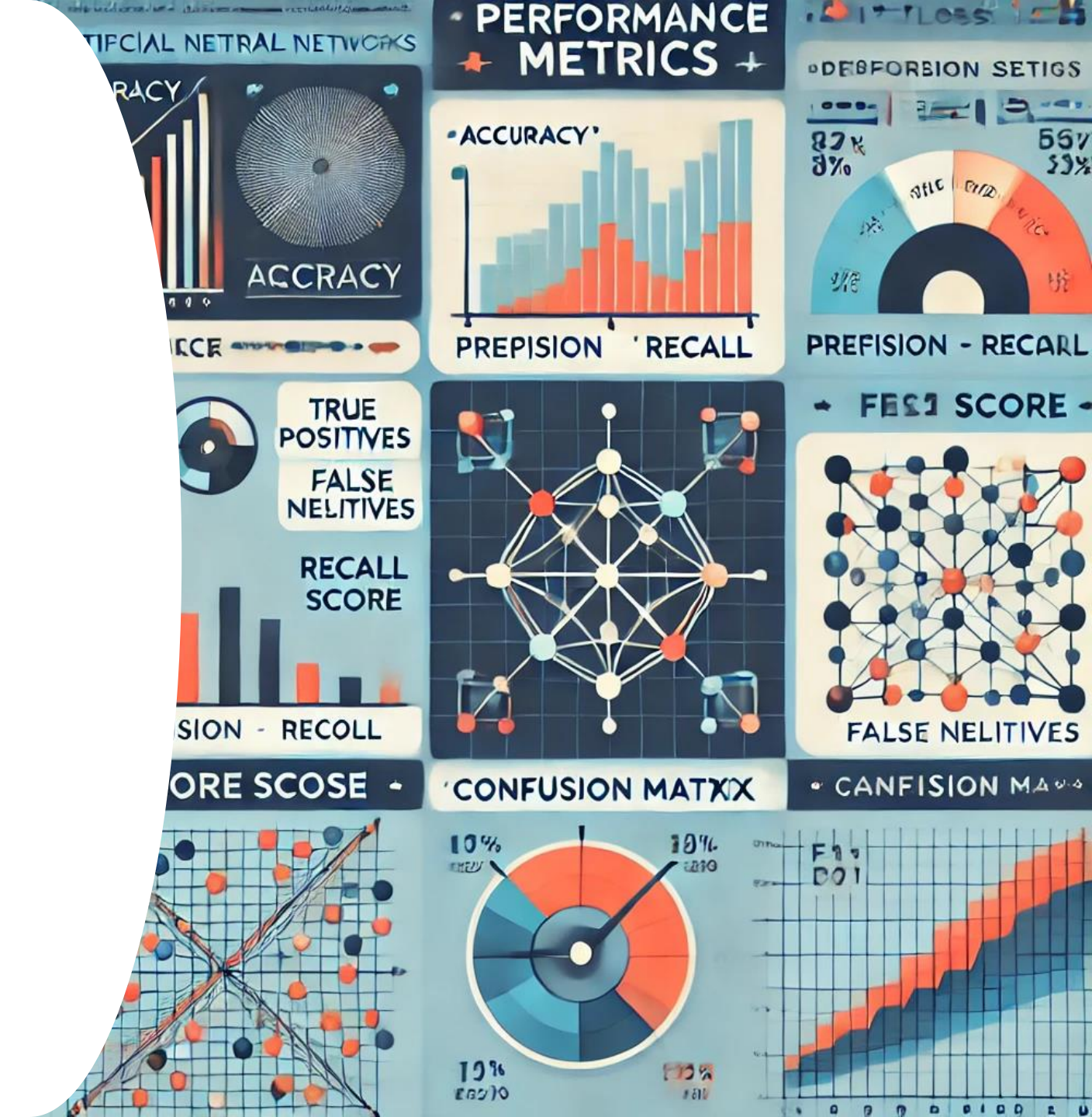
  Test örneği, $Sınıf\ 1$ olarak atanır (çoğunluk oyu).

4. Eşitlik Durumu (Tiebreaker)

Eğer iki veya daha fazla sınıf aynı sayıda komşuya sahipse, bazı stratejiler kullanılabilir:

- Mesafeye göre ağırlıklı oylama (Daha yakın olan komşuların ağırlığı daha fazladır).
- Rastgele bir sınıf seçimi.
- Daha düşük sınıf indeksine sahip sınıfı seçme (örneğin, $sınıf\ 1\ >\ sınıf\ 2$).

Rasgele Orman



4. Rasgele Orman Algoritması

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

Rasgele Orman algoritması, **makine öğrenmesi**nde hem sınıflandırma hem de regresyon problemleri için kullanılan, güçlü ve esnek bir **topluluk öğrenme** (ensemble learning) yöntemidir.

Bu algoritma, birden fazla **Karar Ağacı** (Decision Tree) modelini bir araya getirerek tahminlerin doğruluğunu artırır ve aşırı öğrenmeyi önler.

Özellikleri:

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

- **Topluluk Yaklaşımı:** Birden fazla karar ağacı kullanıldığı için daha kararlı ve doğru tahminler yapılır.
- **Genelleştirme Yeteneği:** Aşırı öğrenmeye karşı dayanıklıdır çünkü tüm ağaçlar farklı veri ve öznitelik kombinasyonları ile eğitilir.
- **Rastgelelik:** Hem veri örneklemesi hem de öznitelik seçimi sırasında rastgelelik kullanılarak ağaçların çeşitliliği sağlanır.

Avantaj ve Dezavantajları:

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

Avantajları:

- Yüksek doğruluk sağlar.
- Hem sınıflandırma hem de regresyon problemlerinde kullanılabilir.
- Eksik veriyle çalışabilir.
- Büyük veri kümeleri ve yüksek boyutlu verilerle iyi performans gösterir.
- Öznitelik önemini (feature importance) belirleyebilir.

Dezavantajları:

- Çok sayıda karar ağacı olduğunda daha fazla hesaplama gücüne ihtiyaç duyar.
- Model yorumlanabilirliği, tek bir karar ağacına göre daha zordur.
- Bellek kullanımı yüksek olabilir.

Çalışma Prensibi:

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

Veri Örnekleme (Bootstrap Sampling):

- Eğitim verisi, tekrar seçime izin vererek (bagging yöntemi) birden fazla alt küme oluşturulur.
- Her alt küme, ana veri setinden rastgele seçilen örneklerden oluşur.

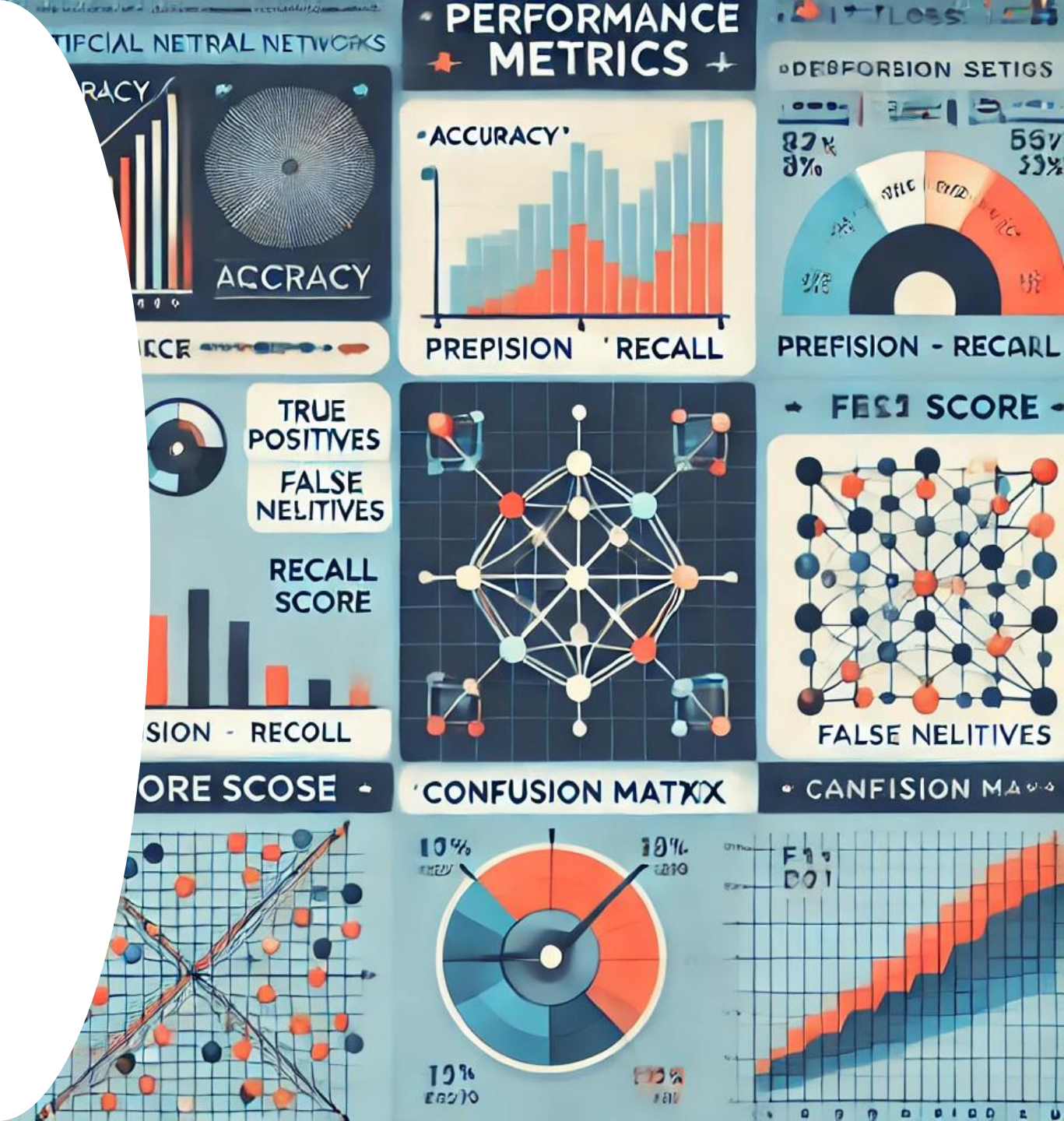
Karar Ağaçlarının Oluşturulması:

- Her alt küme, bir karar ağacı modeli oluşturmak için kullanılır.
- Her bir karar ağacı, **rastgele seçilmiş öznitelik alt kümeleri** kullanılarak eğitilir. Bu özellik, ağaçların çeşitliliğini artırır.

Sonuçların Birleştirilmesi:

- **Sınıflandırma problemlerinde:** Karar ağaçlarının tahmin ettiği sınıflar arasında çoğunluk oyu alınır.

Naive Bayes


PERFORMANCE METRICS
ACCURACY
PREPISION
RECALL
TRUE POSITIVES
FALSE NELITIVES
RECALL SCORE
CONFUSION MATXX
FALSE NELITIVES

5. Naive Bayes (NB):

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

- Olasılık kuramı içinde incelenen bir konudur.
- İstatistiksel bir sınıflandırma yöntemidir ve **Bayes** teoremine dayanır.
- İsmindeki "**Naive**" kısmı, algoritmanın özellikler (öz nitelikler) arasındaki bağımsızlık varsayımını ifade eder.
- Rastlantısal değişken için olasılık dağılım içerisinde şartlı olasılıklar arasındaki ilişkiyi göstermeye çalışır.
- Olasılık teorisi içinde incelenen bir olay olarak B olayına şartlı bir A olayı (yani B olayı bilindiği halde A olayı) için olasılık değeri , A olayına şartlı olarak B olayı (yani A olayı bilindiği halde B olayı) için olasılık değerinden farklıdır.
- Ancak bu iki birbirine ters şartlılık arasında belirli bir ilişki vardır ve buna **Bayes Teoremi** denir.

Naive Bayes (NB):

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

- Ulaşılmak istenilen hedef değişken ile elimizdeki bağımsız değişkenler arasındaki ilişkiyi tespit etmeye çalışan tahmin edici ve tanımlayıcı bir sınıflandırma algoritmasıdır.

- Bu algoritmada, modelin öğrenilmesi sayesinde her çıktının öğrenme kümesinde kaç kere meydana geldiğinin hesaplanmasını ve hesaplanan bu değerlerin **öncelikli olasılık** olarak adlandırılmasını sağlar.

- Bir diğer özelliği de her bağımsız değişken/bağımlı değişken kombinasyonunun oluşma sıklığını da bulmaya çalışmasıdır. Bu sıklıkları tahmin etmek için öncelikli olasılıklar birleştirilir.

- Büyük boyutlu verilerle çalışmada iyi sonuçlar üretemeyebilir.

Bayes Teoremi:

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

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

Burada:

- $P(C|X)$: X özelliklerine sahip bir verinin C sınıfına ait olma olasılığı (posterior probability).
- $P(X|C)$: C sınıfının X özelliklerini üretme olasılığı (likelihood).
- $P(C)$: C sınıfının genel olasılığı (prior probability).
- $P(X)$: X özelliklerinin genel olasılığı (evidence).

Naive Bayes, bu teoremi kullanarak bir verinin hangi sınıfa ait olduğunu tahmin eder.

1.Eğitim Aşaması:

- Verilen veri setinde her sınıfın $P(C)$ (öncül olasılık) değerini hesaplar.
- Her bir özellik için $P(X \mid C)$ değerlerini belirler. Sürekli veriler için genellikle bir dağılım modeli (örneğin, Gaussian dağılımı) kullanılır; kategorik verilerde ise özellik değerlerinin frekansına dayalı olasılıklar hesaplanır.

2.Tahmin Aşaması:

- Yeni bir verinin $P(C \mid X)$ olasılıklarını tüm sınıflar için hesaplar.
- En yüksek olasılığa sahip sınıfı seçerek tahmin yapar.

Naive Bayes Algoritmasının Türleri

1. **Gaussian Naive Bayes:** Sürekli veri ile çalışır ve verilerin normal dağılıma uyduğu varsayılır.
2. **Multinomial Naive Bayes:** Özellikle metin sınıflandırma gibi çok kategorili özelliklere sahip veri setleri için kullanılır.
3. **Bernoulli Naive Bayes:** İkili (binary) özelliklere sahip veriler için uygundur.

Naive Varsayımın Etkisi

- Naive varsayımın etkisi, Naive Bayes algoritmasının temelinde yatan bağımsızlık varsayımından kaynaklanır.
- Bu varsayım, her özelliğin diğerlerinden bağımsız olduğunu kabul eder. Ancak, gerçek dünyada özellikler genellikle birbiriyle ilişkili olur.
- Örneğin, bir üretim hattında bir sensörün ölçtüğü sıcaklık ile bir başka sensörün ölçtüğü titreşim seviyesi arasında bir bağlantı olabilir.

Naive Varsayımın Etkileri

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

1.Hatalı Olasılık Hesaplama:

- Eğer özellikler bağımlıysa (örneğin, sıcaklık arttıkça titreşim de artıyorsa) ve bağımlılık dikkate alınmazsa, model bu ilişkileri görmezden gelir.

Bu da yanlış sınıflandırmaya yol açabilir.

2.Hız ve Basitlik:

- Naive varsayım, hesaplama sürecini basitleştirir ve modeli hızlı hale getirir. Ancak bu hız, doğruluğu negatif etkileyebilir.

3.Çelişkili Sonuçlar:

- Özelliklerin bağımlı olduğu bir durumda, Naive Bayes tahminlerinin gerçek durumdan sapma ihtimali artar.

Etkinin Azaltılması

1. **Özellik Mühendisliği:**

   Özellikler arasındaki ilişkileri ortaya çıkarmak için analiz yapabilir ve bu ilişkilerden yeni özellikler türetilebilir.

2. **Karmaşık Modeller:**

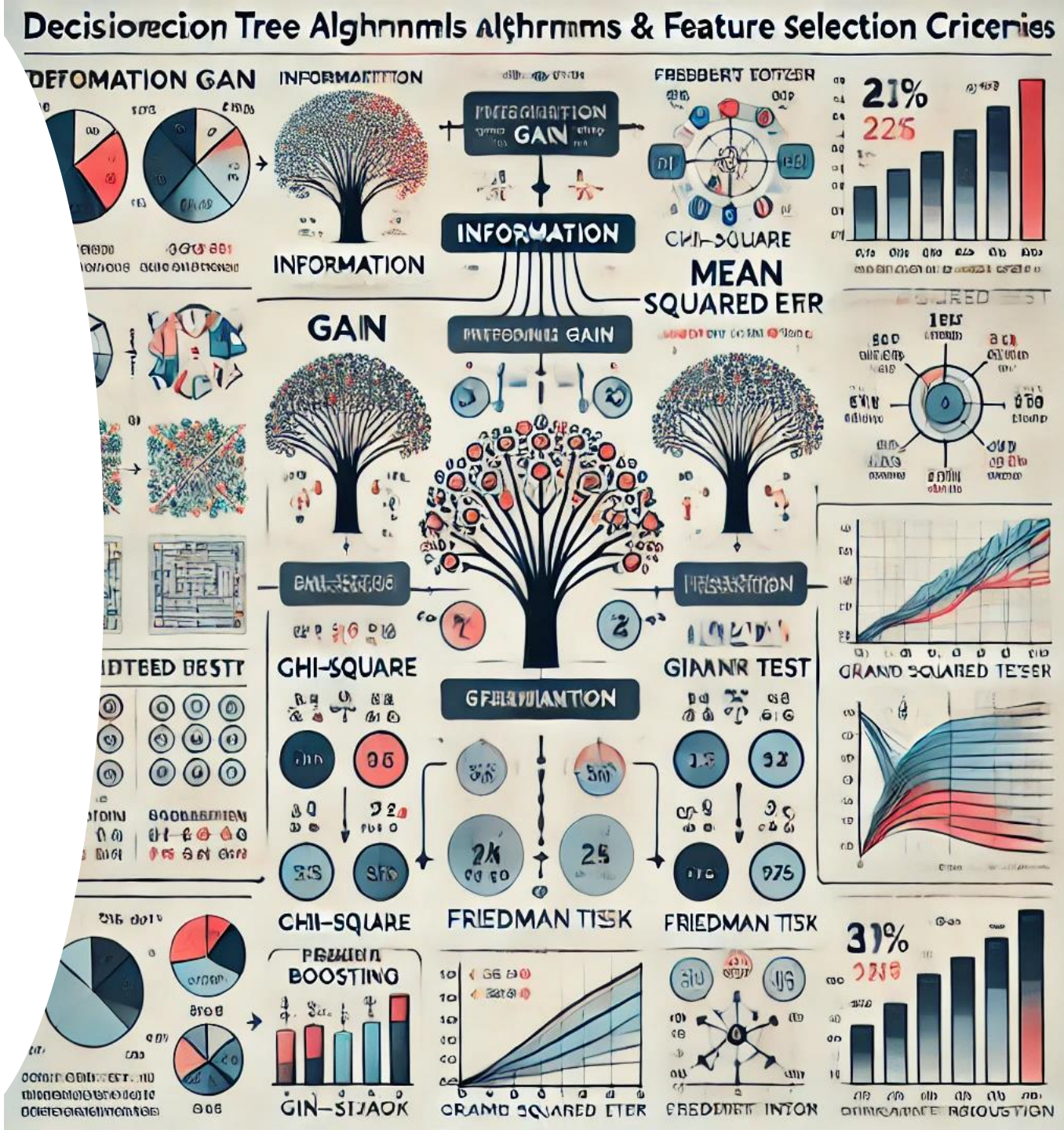
   Daha gelişmiş modeller (örneğin, Karar Ağaçları veya Sinir Ağları) bağımlılıkları öğrenebilir.

3. **Hibrit Yöntemler:**

   Naive Bayes'i başka bir algoritmayla birleştirmek, bağımlılıkların hesaba katılmasını sağlayabilir.

Naive Bayes ile Servo Motor Kontrol Problemi

Karar Ağaçları



6. Karar Ağaçları

- Bir veri setini sınıflandırmak veya tahmin yapmak için kullanılır.
- Verileri basit bir şekilde bölerek, hiyerarşik bir yapı oluşturur ve sonuçta **"ağaç" yapısına** benzer bir model elde edilir.

Temel Kavramlar

1.Kök Düğüm (Root Node):

- Karar ağacının başlangıç noktasıdır. Bu düğüm, veri setindeki en iyi özelliğe göre bölünmeyi temsil eder.

2.Dahili Düğüm (Internal Node):

- Verinin belirli bir özelliğe göre bölündüğü düğümlerdir. Her dahili düğüm, bir özelliği ve bu özelliğin değerlerini kullanarak kararlar alır.

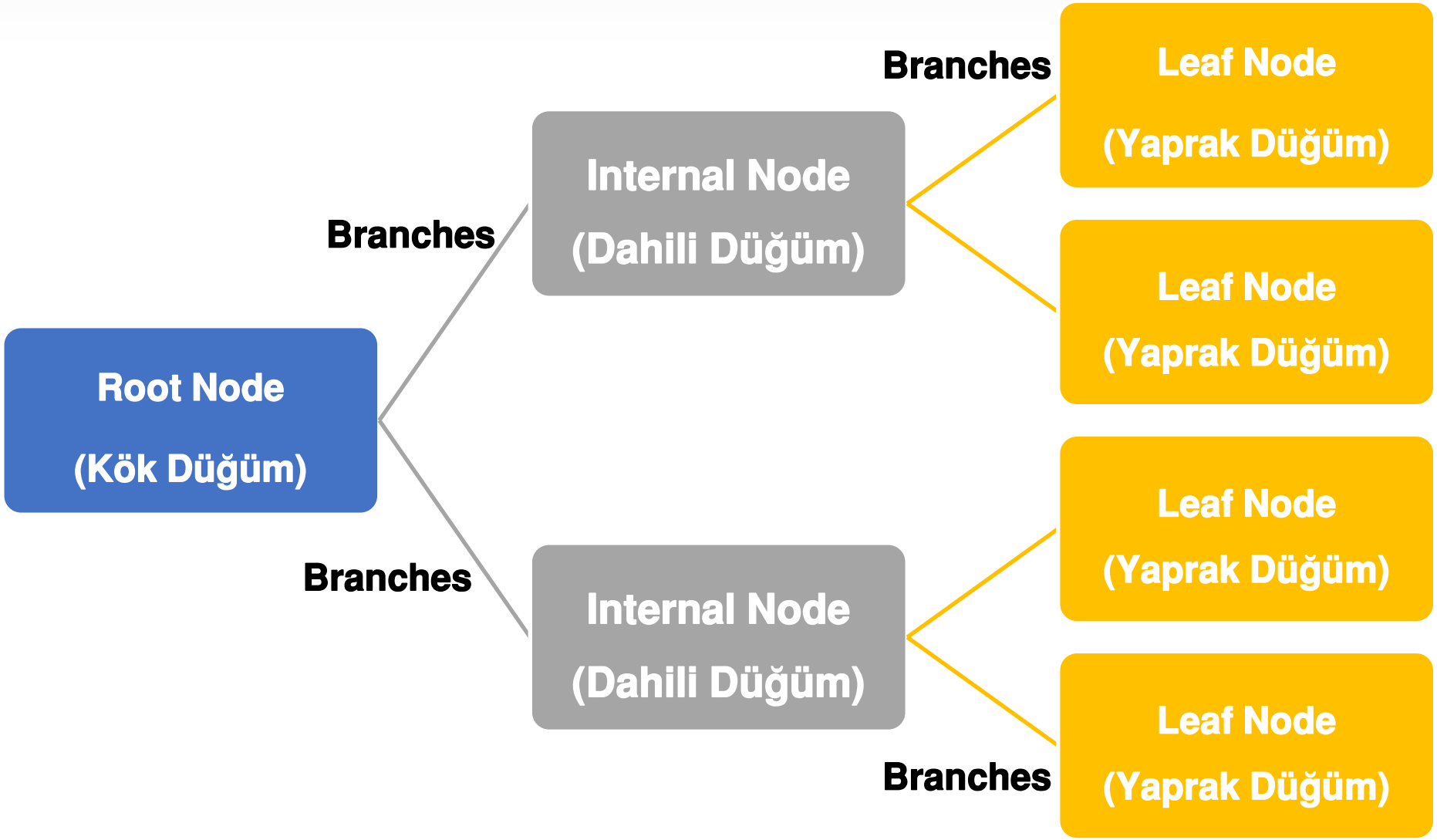
3.Yaprak Düğüm (Leaf Node):

- Karar ağacının son düğümleridir. Bu düğümler, bir sınıf veya tahmin değerini temsil eder.

4.Dallanma (Branch):

- Bir düğümden diğerine giden yolları ifade eder. Bu yollar, belirli bir özelliğin değerine bağlıdır.

Karar Ağaç Yapısı



Temel Çalışma Prensibi

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

1. **Özellik Seçimi:**

- Ağaç, veriyi bölebilmek için bir özellik seçer. Özellik seçimi, genellikle **bilgi kazancı (information gain)** veya **Gini indeksine** dayanır.
- Örnek: «Konum», »Kuvvet», »Basınç» gibi özellikler.

2. Bölünme:

- Veri, seçilen özelliğin belirli değerlerine göre dallara ayrılır.
- Örnek: »Kuvvet > 5 N mu?» sorusuna "Evet" ve "Hayır" cevaplarıyla ikiye ayrılabilir.

3. Tekrarlama:

- Bölünme işlemi, her alt grupta en iyi özellik seçilerek tekrarlanır.

4. Durdurma Kriteri:

- Tüm veriler aynı sınıfa ait olduğunda veya belirli bir derinlik sınırına ulaşıldığında ağaç genişlemesi durur.

Örnek: Robot Kol Tipi Seçim Örneği

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

Karar Ağacının Yapısı:

1. **Kök Düğüm (Root Node):**

"Yük kapasitesi ≥ 10 kg mı?"

2. Dahili Düğümler (Internal Nodes):

Çalışma Alanı ≥ 1.5 metre mi?

3.Yaprak Düğümler (Leaf Nodes):

Robot Tipleri

Endüstriyel Robot: Yüksek yük ve geniş hareket alanı.

Hassas Robot: Hassas ve küçük işler için uygun.

Hızlı Robot: Daha düşük yükle yüksek hız.

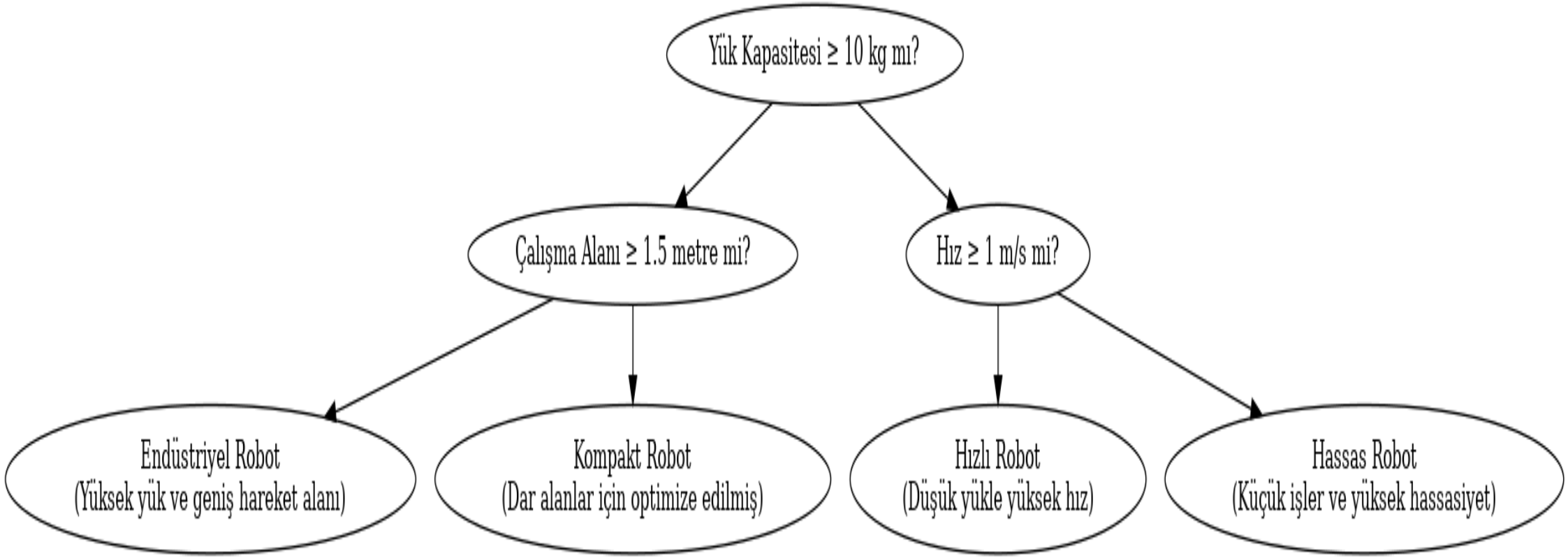
Kompakt Robot: Dar alanlar için optimize edilmiş.

Örnek: Robot Kol Tipi Seçim Örneği

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

Düğüm Türü	Karar Kriteri	Sonuç
Kök Düğüm	$Yük\ Kapasitesi \geq 10\ kg$ mı?	**Evet** → Daha güçlü bir robot kol seç. **Hayır** → Hafif işler için uygun bir robot kol seç.
Dahili Düğüm 1	$Hız \geq 1\ m/s$ mi?	**Evet** → Hızlı ve orta yük taşıyan robot kol seç. **Hayır** → Yavaş ama hassas robot kol seç.
Dahili Düğüm 2	$Çalışma\ Alanı \geq 1.5\ metre$ mi?	**Evet** → Geniş hareket alanı için endüstriyel robot seç. **Hayır** → Kompakt robot seç.
Yaprak Düğüm 1	Endüstriyel Robot	Yüksek yük kapasitesi ve geniş hareket alanı.
Yaprak Düğüm 2	Hassas Robot	Küçük işler ve yüksek hassasiyet gerektiren uygulamalar.
Yaprak Düğüm 3	Hızlı Robot	Düşük yük gereksinimleriyle yüksek hızda çalışabilen robotlar.
Yaprak Düğüm 4	Kompakt Robot	Dar alanlar için optimize edilmiş robotlar.

Robot Kol Seçim Karar Ağacı Yapısı



Karar Ağaçları Algoritmaları:

Algoritma	Özellik Seçimi Kriteri	Açıklama	Kullanım Durumu
ID3	Bilgi Kazancı (Information Gain)	En yüksek bilgi kazancı sağlayan özelliği seçer. Basit ve hızlıdır ancak sürekli verilerle çalışması zordur.	Küçük ve kategorik veri setlerinde kullanılır. Basit sınıflandırma problemleri için uygundur.
C4.5	Bilgi Kazancı ve Normalizasyon	ID3'ün geliştirilmiş versiyonudur. Sürekli verilerle çalışabilir ve eksik verilerle başa çıkabilir.	Sürekli ve kategorik verilerin bir arada olduğu, daha karmaşık veri setlerinde tercih edilir.
CART	Gini İndeksi veya Ortalama Kare Hata	Hem sınıflandırma hem de regresyon problemleri için uygundur. Karar ağacını sadece ikili dallara ayırır.	Hem sınıflandırma hem de regresyon problemleri için genel amaçlı olarak kullanılır.
CHAID	Ki-kare testi	Özellikler arasındaki bağımsızlığı ölçer. Kategorik verilerle iyi çalışır.	Pazarlama analizleri ve kategorik veri setlerinde kullanılabilir.
Random Forest	Rastgele Alt Kümeler	Birden fazla karar ağacı kullanarak overfitting'i azaltır. Sonuçları oylama veya ortalama ile birleştirir.	Büyük veri setlerinde ve aşırı uydurma riski olan durumlarda yüksek doğruluk elde etmek için uygundur.
XGBoost	Gradient Boosting	Karar ağaçlarını ardışık olarak oluşturur ve önceki ağaçların hatalarını düzeltir. Büyük veri setlerinde yüksek performanslıdır.	Çok büyük veri setleri ve yüksek doğruluk gerektiren problemler için uygundur.
Gradient Boosting Machines (GBM)	Gradient Boosting	Hataları minimize etmek için karar ağaçlarını kademeli olarak iyileştirir.	XGBoost'un alternatifi olarak daha az karmaşık parametre ayarları gerektiren durumlarda tercih edilir.
Hızlı Karar Ağaçları	Optimizasyon Bazlı Bölme	Büyük veri kümelerinde hızlı sonuç üretmek için optimize edilmiştir.	Çok büyük veri setlerinde, hızın daha önemli olduğu durumlarda tercih edilir.

Karar Ağaçları Algoritmalarının Avantaj ve Dezavantajları:

Algoritma	Avantajlar	Dezavantajlar
ID3	- Basit ve hızlıdır.	- Sürekli verilerle çalışması zordur. - Aşırı uydurmaya (overfitting) meyillidir.
C4.5	- Hem sürekli hem de kategorik verilerle çalışabilir. - Eksik verilerle başa çıkabilir.	- Hesaplama daha karmaşıktır. - Çok büyük veri setlerinde yavaş çalışabilir.
CART	- Basit, etkili ve hızlıdır. - Hem sınıflandırma hem de regresyon için uygundur.	- Çok derin ağaçlar oluşturabilir, bu da aşırı öğrenmeye yol açabilir.
CHAID	- Hızlıdır ve kategorik verilerle iyi çalışır.	- Sürekli verilerle çalışması zordur. - Çok sayıda kategorisi olan özelliklerle performansı düşebilir.
Random Forest	- Overfitting'i azaltır. - Yüksek doğruluk sağlar. - Eksik verilerle başa çıkabilir.	- Eğitim süresi uzundur. - Modelin yorumlanabilirliği azdır.
XGBoost	- Hızlıdır ve ölçeklenebilir. - Eksik verilerle başa çıkabilir. - Çok yüksek doğruluk sağlar.	- Parametre ayarları karmaşıktır. - Aşırı optimizasyon aşırı uydurmaya neden olabilir.
Gradient Boosting Machines (GBM)	- Yüksek doğruluk sağlar. - Hataları kademeli olarak azaltır.	- Eğitim süresi uzundur. - Parametre ayarları detaylı bilgi gerektirir.
Hızlı Karar Ağaçları	- Büyük veri kümelerinde hızlı sonuç üretir. - Düşük maliyetlidir.	- Çok karmaşık problemlerle başa çıkmada sınırlıdır.

Özellik Seçim Kriterleri-1:

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS LABORATORY

Kriter	Hesaplama Yöntemi	Denklem	Kullanım Alanı
Bilgi Kazancı (Information Gain)	Entropi ile veri belirsizliğini ölçerek en yüksek bilgi kazancını sağlayan özelliği seçer.	$IG = Entropy(parent) - \sum_{i=1}^{k} \frac{\lvert S_i \rvert}{\lvert S \rvert} \cdot Entropy(S_i)$	
Gini İndeksi (Gini Index)	Veriyi sınıflara ayırırken saflığı (purity) artıran özelliği seçer.	$Gini = 1 - \sum_{i=1}^{k} p_i^2$, burada p_i sınıf i'nin oranıdır.	Sınıflandırma problemleri için uygundur. Çoklu sınıf problemlerinde sıkça tercih edilir.
Ki-Kare (Chi-Square)	Özellikler arasındaki bağımsızlığı ölçer ve en yüksek bağımlılığı sağlayan özelliği seçer.	$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$, burada O_i gözlenen değer ve E_i beklenen değerdir.	Kategorik verilerle çalışırken, özellikle pazarlama ve müşteri analizlerinde kullanılır.
Ortalama Kare Hata (Mean Squared Error)	Sürekli hedef değişkenlerin tahmini için kare hatalarını minimize eden özelliği seçer.	$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$, burada y_i gerçek değer ve $\hat{y}_i$ tahmin edilen değerdir.	Regresyon problemlerinde kullanılır.

IG: Bilgi kazancı

$Entropy(parent)$: Tüm veri setinin entropisi

S_i: Alt grubun eleman sayısı

S: Tüm veri setinin eleman sayısı

$Entropy(S_i)$: Alt grubun entropisi

$$Entropy(parent) = -\sum_{i=1}^{k} p_i \cdot \log_2(p_i)$$

k: Veri setindeki sınıf sayısı.

p_i: Sınıf i'nin veri setindeki olasılığı ($p_i = \frac{\text{sınıf i'nin eleman sayısı}}{\text{toplam eleman sayısı}}$).

Özellik Seçim Kriterleri-2:

Gain Ratio (Kazanç Oranı)	Bilgi kazancını normalleştirerek dengesiz bölünmeleri önler.	$GR = \frac{InformationGain}{SplitInformation}$, burada $SplitInformation = -\sum_{i=1}^{n} p_i \cdot \log_2(p_i)$.	Sürekli ve kategorik verilerin bir arada olduğu durumlarda tercih edilir.
Friedman Testi	Farklı özelliklerin performansını kıyaslar ve önemli özellikleri belirler.	$\chi_F^2 = \frac{12}{n \cdot k(k+1)} \sum_{j=1}^{k} R_j^2 - 3n(k+1)$, burada R_j toplam sıralamadır.	Büyük veri setlerinde, özellikle regresyon problemlerinde kullanılır.
Gradient Boosting	Gradient Descent yöntemini kullanarak hata oranını minimize eden özelliği seçer.	$Gradient = -\frac{\partial Loss}{\partial Prediction}$, burada $Loss$ kayıp fonksiyonudur.	Boosting algoritmalarında kullanılır. Büyük veri setlerinde doğruluğu artırmak için uygundur.
Variance Reduction (Varyans Azaltma)	Hedef değişkenin varyansını minimize eden özelliği seçer.	$Variance\ Reduction = Variance(parent) - \sum_{i=1}^{k} \frac{\lvert S_i \rvert}{\lvert S \rvert} \cdot Variance(S_i)$	

$Variance(parent)$: Ana veri kümesinin varyansı

$\frac{|S_i|}{|S|}$: Alt grubun tüm veri setine oranı

$Variance(S_i)$: Alt grubun varyansı

ID3 (Iterative Dichotomiser 3) Algoritması:

- **Ross Quinlan** tarafından geliştirilmiş bir karar ağacı oluşturma algoritmasıdır.
- Bu algoritma, karar ağacının dallarını oluştururken **Bilgi Kazancı (Information Gain)** kavramını kullanır.
- **Temel amacı**, veri setindeki belirsizliği azaltarak veriyi sınıflandırmak için en uygun özellikleri seçmektir.
- "Dichotomiser" kelimesi, "ikiye ayıran" anlamına gelir ve genellikle karar ağacı algoritmaları bağlamında kullanılır. Bu terim, veri setini farklı alt gruplara ayırma işlemini ifade eder. ID3 algoritmasında, her adımda bir özellik seçilerek veri ikiye ayrılır ve bu işlem, karar ağacı yapısını oluşturur.

Algoritmanın Çalışma Prensibi

1. **Kök Düğümün Belirlenmesi:**
   - Veri setindeki en yüksek **Bilgi Kazancı** sağlayan özellik seçilir ve kök düğüm olarak atanır.
2. **Veriyi Bölme:**
   - Kök düğümde seçilen özelliğin her bir değeri için yeni alt gruplar oluşturulur.
3. **Tekrarlama:**
   - Alt gruplar için tekrar en yüksek bilgi kazancına sahip özellikler belirlenir ve dallar oluşturulur.
4. **Durdurma Kriteri:**
   - Tüm veriler aynı sınıfa ait olduğunda veya kullanılabilecek özellik kalmadığında süreç durur.

Entropi ve Bilgi Kazancı

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

Entropi:

- Entropi, veri setindeki belirsizliği ölçer.

$$Entropy(S) = -\sum_{i=1}^{n} p_i \cdot \log_2(p_i)$$

Burada:

- p_i: i sınıfının oranı
- S: Veri seti

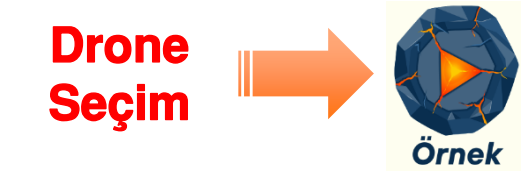
Bilgi Kazancı:

- Her bir özellik için bilgi kazancı şu şekilde hesaplanır:

$$IG(A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \cdot Entropy(S_v)$$

Burada:

- A: Özellik
- S_v: A özelliğinin v değerine sahip olan alt veri seti



C4.5 Algoritması

- C4.5 algoritması, karar ağaçları (decision trees) oluşturmak için kullanılan ve makine öğrenmesi alanında popüler bir algoritmadır.
- Ross Quinlan tarafından geliştirilmiş olup, ID3 algoritmasının bir geliştirilmiş versiyonudur.
- C4.5, sınıflandırma problemlerinde verilerin belirli sınıflara atanması için etkili bir yöntem sunar.
- C4.5 isminin kaynağı, Ross Quinlan'ın daha önce geliştirdiği **ID3 (Iterative Dichotomiser 3)** algoritmasından gelmektedir. C4.5, ID3 algoritmasının bir geliştirilmiş sürümü olduğu için, bu isimlendirme bir devam niteliği taşır

C4.5 Algoritmasının Temel Özellikleri

- **Karar Ağacı Oluşturma**: C4.5, eğitim veri setindeki örneklerden bir karar ağacı oluşturur. Bu ağaç, veri örneklerini belirli bir sınıfa atamak için kurallar içerir.
- **Entropi ve Bilgi Kazancı**: Algoritma, her bir özelliğin sınıflandırmaya olan katkısını ölçmek için **bilgi kazancı oranını (gain ratio)** kullanır. Bu, özelliklerin önemini belirler ve en uygun özelliği seçer.
- **Sürekli ve Kategorik Verilerle Çalışma**: C4.5, hem sürekli (numerik) hem de kategorik verilerle çalışabilir. Sürekli değerler için eşik belirleyerek bu değerleri kategorik hale getirir.
- **Eksik Verilerle Başa Çıkma**: Eksik değerlere sahip özellikler için, C4.5 bu verilerin etkisini minimize edecek yöntemler uygular.
- **Pruning (Budama)**: Karar ağacındaki aşırı karmaşıklığı önlemek için **post-pruning** işlemi gerçekleştirir. Bu işlem, aşırı öğrenmenin (overfitting) önüne geçer.

C4.5 Algoritmasının Çalışma Prensibi

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

1.Kök Düğümün Seçilmesi:

- Algoritma, tüm özellikleri değerlendirerek, bilgi kazancı oranı en yüksek olan özelliği kök düğüm olarak seçer.
- Bilgi kazancı, bir özelliğin veri örneklerini sınıflar arasında ne kadar iyi ayırabildiğini ölçer.

2.Alt Ağaçların Oluşturulması:

- Her özelliğin değerlerine göre veri seti dallara ayrılır.
- Her bir alt dal için aynı işlem tekrarlanarak alt ağaçlar oluşturulur.

3.Sürekli Değerlerin İşlenmesi:

- Sürekli özellikler için, veri seti belirli bir eşik değere göre iki gruba ayrılır.

C4.5 Algoritmasının Çalışma Prensibi

4. Eksik Verilerin Yönetimi:

- Eksik değerlere sahip veriler, olasılıksal yöntemlerle dallara atanır veya bilgi kazancı hesaplamasına dahil edilmez.

5. Karar Ağacının Sonuçlandırılması:

- Tüm veri örnekleri sınıflandırıldığında veya daha fazla bilgi kazancı sağlanamadığında ağaç sonlandırılır.

6. Ağacın Budanması (Pruning):

- Karar ağacındaki gereksiz dallar budanır. Bu işlem, ağaç performansını artırır ve overfitting'i azaltır.

- C4.5, yerini J48 gibi modern varyasyonlara bıraksa da, makine öğrenmesi ve karar ağacı algoritmalarının gelişiminde önemli bir kilometre taşıdır.
- Modern uygulamalarda genellikle **Random Forest** veya **Gradient Boosting Trees** gibi daha gelişmiş algoritmalar kullanılır.
- Ancak temel sınıflandırma problemlerinde C4.5 hâlâ etkili bir yöntemdir

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

TEŞEKKÜRLER

Prof. Dr. Erhan AKDOĞAN

eakdogan@yildiz.edu.tr

www.biomech.yildiz.edu.tr

6. Karar Ağaçları

YILDIZ TECHNICAL UNIVERSITY
BIOMECHATRONICS
LABORATORY

- Çok sayıda veri içeren bir veri tabanını bazı teknikler kullanarak alt bölümlere ayırmak için geliştirilmiştir.
- Sınıf seçenekleri ile olasılıklara bağlı durumları belirli bir düzen içinde sıralar.
- Kategorik ve nümerik veriler üzerinde çalışabilir.
- İki aşamada çalışır: öğrenme, uygulama.

- Kayıt kök düğümden başlar ve ara düğümlerden hangi yöne dallanacağı belirlenir.
- Her bir sınıf ağaçta tek yaprak olarak gösterilir.
- Bu yüzden bir sınıfa giden yol yalnızca bir tane olmalıdır.
- Yapraklar arasında herhangi kısa bir yol yoktur.
- Dallanma işlemi yaprak düğüme ulaşıncaya kadar devam eder.