1. Sınıflandırma Nedir?

Sınıflandırma, makine öğrenmesinin denetimli öğrenme kategorisinde yer alan bir problemdir. Bu işlemde amacımız, verilerdeki örnekleri belirli sınıflara ayırmaktır. **Denetimli öğrenme**, etiketli veriler kullanılarak modelin eğitilmesidir.

Örnek: Bir e-posta sınıflandırma sisteminde, e-postaların "spam" mı yoksa "spam değil" mi olduğunu belirlemek.

Sınıflandırmanın Gerçek Dünya Uygulamaları:

- **Sağlık:** Kanserli hücreleri tanımak için biyomedikal verilerle yapılan sınıflandırmalar.
- **E-ticaret:** Kullanıcılara ürün önerileri.
- **Güvenlik:** Yüz tanıma veya plaka okuma gibi uygulamalar.

2. Binary ve Çoklu Sınıflandırma

- **Binary (İkili) Sınıflandırma:** Veriyi iki sınıfa ayırmaktır. Örneğin, e-posta “spam” veya “spam değil” gibi. Burada yalnızca iki olasılık vardır.
- **Çoklu Sınıflandırma:** Veriyi üç veya daha fazla sınıfa ayırmaktır. Örneğin, el yazısı tanıma sisteminde, rakamların 0’dan 9’a kadar sınıflandırılması.

3. One-vs-Rest (OvR) ve One-vs-One (OvO) Stratejileri

Çoklu sınıflandırma için **OvR** ve **OvO** gibi iki ana strateji bulunur.

- **One-vs-Rest (OvR):** Her bir sınıf için ayrı bir model eğitilir ve her model "bu sınıf" ve "diğer sınıflar" arasında ayrım yapar. Bu strateji genellikle daha hızlıdır ancak **dengesiz veri kümelerinde** zayıf performans gösterebilir.
- **One-vs-One (OvO):** Her bir sınıf çifti için ayrı bir model eğitilir. K sınıfı için toplamda **K(K-1)/2** model oluşturulur. Bu strateji genellikle daha iyi sonuç verir ama daha fazla hesaplama gücü gerektirir.

4. Destek Vektör Makineleri (SVM)

SVM, **hem sınıflandırma hem de regresyon problemleri** için kullanılabilen güçlü bir algoritmadır. Ancak özellikle sınıflandırma için yaygın olarak kullanılır. SVM’nin temel amacı, veriyi **en iyi şekilde ayıran hiperdüzlemi** (decision boundary) bulmaktır.

- **Marjin:** SVM, sınıfları ayıran **hiperdüzlem** ile sınıfların en yakın veri noktaları arasındaki mesafeyi en büyük yapmak ister. Bu mesafe "marjin" olarak adlandırılır.

- **Doğrusal Sınıflandırma:** Eğer veriler doğrusal olarak ayrılabiliyorsa, SVM, verileri ayırmak için doğrusal bir hiperdüzlem bulur. Yani, veriler tek bir düz çizgi ile sınıflandırılabilir.
- **Doğrusal Olmayan Sınıflandırma:** Çoğu veri genellikle doğrusal olmayan şekilde dağılır. Bu durumda, **kernel** fonksiyonları kullanılır. Bu fonksiyonlar verileri daha yüksek boyutlu bir uzaya dönüştürür, böylece doğrusal bir sınıflandırma yapılabilir.

5. Kernel Fonksiyonları

Kernel, iki veri noktası arasındaki benzerliği ölçen bir fonksiyondur. Kernel fonksiyonları, veriyi doğrusal olmayan uzaylarda ayrılabilir hale getirebilir.

- **Doğrusal Kernel:** Basit veriler için kullanılır, doğrusal olarak ayrılabilen verilerde etkilidir.
- **RBF Kernel (Radial Basis Function):** Genellikle daha karmaşık veri kümelerinde kullanılır. Çoğu zaman en iyi sonucu verir.

6. Hiperparametreler

SVM'nin başarısı, kullanılan hiperparametrelerin doğru şekilde ayarlanmasına bağlıdır:

- **C (Ceza Parametresi):**
    - Yüksek C, modelin marjin hatalarına karşı daha toleranssız olmasını sağlar, ancak aşırı uyuma (overfitting) yol açabilir.
    - Düşük C, daha fazla hata kabul eder, ancak model daha genel olabilir.
- **Gamma:** Özellikle **RBF kernel** kullanıldığında, modelin karmaşıklığını kontrol eder. Gamma değeri küçükse, model daha basit ve düz sınırlar oluşturur; büyükse, daha karmaşık sınırlar çizer.

7. SVM'nin Avantajları ve Dezavantajları

Avantajları:

- **Genelleme Yeteneği:** SVM, marjini maksimize ederek modelin genelleme yeteneğini artırır.
- **Aykırı Değerlere Karşı Dayanıklı:** Aykırı veriler SVM tarafından iyi bir şekilde işlenir.
- **Yüksek Boyutlu Verilerle Etkili:** Çok boyutlu veri setlerinde güçlü performans gösterir.

Dezavantajları:

- **Büyük Veri Setlerinde Yavaşlık:** SVM, büyük veri setlerinde eğitim süreci açısından zaman alıcı olabilir.
- **Hiperparametre Seçimi Zorluğu:** C ve gamma gibi hiperparametrelerin doğru ayarlanması zor olabilir.
- **Dengesiz Veri Kümeleri:** Eğer sınıflar arasında büyük bir dengesizlik varsa, SVM bazen iyi sonuç vermez.

8. Kernel Seçimi

Kernel seçiminde:

- Eğer veri **doğrusal olarak ayrılabiliyorsa**, **doğrusal kernel** tercih edilir.
- Eğer veri **doğrusal olarak ayrılmıyorsa**, **RBF kernel** daha iyi sonuç verebilir.

Aşağıdaki hangi durum binary (ikili) sınıflandırma problemidir?

A) Bir meyvenin elma mı yoksa muz mu olduğunu tahmin etmek
B) Bir e-posta mesajının "spam" mı yoksa "spam değil" mi olduğunu belirlemek
C) Bir kişinin yaşadığı şehirlerin listesini oluşturmak
D) 0'dan 9'a kadar olan rakamları tanımak

Doğru Cevap:
B) Bir e-posta mesajının "spam" mı yoksa "spam değil" mi olduğunu belirlemek

Açıklama: Binary (ikili) sınıflandırma, yalnızca iki sınıf arasında seçim yapmayı gerektirir. "Spam" ve "spam değil" iki sınıf olduğundan bu örnek binary sınıflandırma problemidir. Diğer seçeneklerde birden fazla sınıf yer almaktadır.

---

2. Çoklu sınıflandırma problemi için aşağıdaki stratejilerden hangisi doğru değildir?

A) One-vs-Rest (OvR) her sınıf için ayrı bir model oluşturur.
B) One-vs-One (OvO) her sınıf çifti için ayrı bir model oluşturur.
C) OvR, her modelin "bu sınıf" ve "diğer tüm sınıflar" arasında ayrım yapmasını sağlar.
D) OvO, her modelin bir sınıfı bir diğer sınıf ile karşılaştırmasını sağlar.

Doğru Cevap:
D) OvO, her modelin bir sınıfı bir diğer sınıf ile karşılaştırmasını sağlar.

Açıklama: One-vs-One (OvO) stratejisinde, her sınıf çifti için ayrı bir model eğitilir. Ancak bu, her modelin yalnızca iki sınıfı karşılaştırması gerektiği anlamına gelir. Yani her model yalnızca ilgili iki sınıfı dikkate alır, diğer sınıflar göz ardı edilir.

---

3. SVM'nin amacı nedir?

A) Veriyi doğru sınıfa atamak
B) Verileri en büyük marjinle ayıran bir hiperdüzlem bulmak
C) Verileri doğrusal bir şekilde sınıflandırmak
D) Verileri olabildiğince fazla sayıda sınıfa ayırmak

Doğru Cevap:
B) Verileri en büyük marjinle ayıran bir hiperdüzlem bulmak

Açıklama: SVM'nin temel amacı, verileri **en büyük marjinle** ayıran bir **hiperdüzlem** (decision boundary) bulmaktır. Marjin, sınıflar arasındaki en yakın noktalar (destek vektörleri) ile hiperdüzlem arasındaki mesafedir. Bu, SVM'nin genel doğruluğunu artırmaya yardımcı olur.

---

4. SVM'nin kullanımı için hangi durum geçerli değildir?

A) Veriler doğrusal olarak ayrılabiliyorsa
B) Veriler doğrusal olarak ayrılmıyorsa ve daha karmaşık sınırlar gerekiyor
C) Veriler çok büyükse
D) Veriler çok küçükse ve sınıflar arasında belirgin farklar varsa

Doğru Cevap:
C) Veriler çok büyükse

Açıklama: SVM, büyük veri setlerinde **eğitim süresi** açısından yavaş olabilir ve **hesaplama maliyeti** yüksektir. Küçük ve orta ölçekli veri kümelerinde oldukça başarılıdır, ancak büyük veri kümeleriyle başa çıkmak daha zor olabilir.

---

5. Aşağıdaki hangi kernel türü, doğrusal olmayan verileri doğru sınıflandırmak için en yaygın kullanılan kernel'dir?

A) Polinomsal kernel
B) RBF kernel

C) Doğrusal kernel
D) Fourier kernel

Doğru Cevap:
B) RBF kernel

Açıklama: RBF (Radial Basis Function) kernel, doğrusal olmayan verilerle çalışmak için **en yaygın kullanılan kernel türüdür**. Verileri daha yüksek boyutlu bir uzaya dönüştürür, böylece doğrusal olmayan veriler doğrusal hale gelir ve ayrılabilir.

Destek Vektör Makineleri (SVM) için aşağıdaki ifadelerden hangisi doğrudur?

A) SVM yalnızca doğrusal sınıflandırma problemleri için uygundur.
B) SVM, verileri doğru sınıfla en iyi şekilde ayırmak için en küçük marjini tercih eder.
C) SVM, doğrusal olmayan sınıflandırma problemleri için kernel fonksiyonları kullanır.
D) SVM, sınıflar arasındaki mesafeyi minimuma indirgemek için tasarlanmıştır.

Doğru Cevap:
C) SVM, doğrusal olmayan sınıflandırma problemleri için kernel fonksiyonları kullanır.

Açıklama: SVM, **kernel fonksiyonları** kullanarak doğrusal olmayan sınıflandırma problemlerini de çözebilir. Bu fonksiyonlar, verileri daha yüksek boyutlu bir uzaya dönüştürerek doğrusal hale getirir.

---

2. SVM'de "marjin" terimi neyi ifade eder?

A) Veriler arasındaki en büyük mesafe
B) Sınıflar arasındaki en yakın veri noktalarıyla hiperdüzlem arasındaki mesafe
C) Verilerin birbirine olan benzerliği
D) Modelin doğruluğunu ölçen bir skorlama sistemidir

Doğru Cevap:
B) Sınıflar arasındaki en yakın veri noktalarıyla hiperdüzlem arasındaki mesafe

Açıklama: Marjin, SVM'nin bulduğu hiperdüzlem ile her sınıfa ait en yakın veri noktaları (destek vektörleri) arasındaki mesafedir. Bu mesafe ne kadar büyük olursa, modelin **genelleme** yeteneği o kadar iyi olur.

---

3. Hangi durumda C (ceza parametresi) değeri düşük olmalıdır?

A) Veri seti çok büyükse
B) Model aşırı uyum yapıyorsa (overfitting)
C) Model düşük doğruluk veriyorsa
D) Model çok fazla hata kabul ediyorsa ve basit sınıflama isteniyorsa

Doğru Cevap:
D) Model çok fazla hata kabul ediyorsa ve basit sınıflama isteniyorsa

Açıklama: C parametresi küçük olduğunda model daha fazla hata kabul eder, bu da **underfitting**'e yol açabilir. Ancak, veri seti karmaşık değilse ve modelin daha genel olmasını istiyorsanız, düşük **C** değeri uygundur.

---

4. RBF kernel'in gamma parametresi hakkında aşağıdaki ifadelerden hangisi doğrudur?

A) Gamma değeri ne kadar küçükse, modelin karmaşıklığı o kadar artar.
B) Gamma değeri yüksek olduğunda, model daha düz ve basit sınırlar çizer.
C) Gamma, her bir örneğin ne kadar uzağa etkili olacağını belirler ve küçük gamma daha genel bir model oluşturur.
D) Gamma parametresi, yalnızca doğrusal kernel fonksiyonları için geçerlidir.

Doğru Cevap:
C) Gamma, her bir örneğin ne kadar uzağa etkili olacağını belirler ve küçük gamma daha genel bir model oluşturur.

Açıklama: Gamma parametresi, RBF kernel kullanıldığında modelin karmaşıklığını kontrol eder. Küçük bir gamma değeri, modelin daha düz sınırlar oluşturmasına neden olur ve daha genel hale gelir. Büyük gamma, daha keskin sınırlar çizer ve modelin daha karmaşık olmasına yol açar.

---

5. Aşağıdaki durumların hangisinde SVM kullanımı daha verimli olabilir?

A) Verilerin birbirine çok yakın olduğu ve sınıfların net bir şekilde ayrılmadığı durumlarda.
B) Aykırı verilerin çok fazla olduğu ve modelin çok fazla hata yaptığı durumlarda.
C) Yüksek boyutlu verilerin olduğu durumlarda, özellikle çok fazla özellik (feature) varsa.
D) Veri setinin çok küçük olduğu ve yalnızca birkaç örnek bulunduğu durumlarda.

Doğru Cevap:
C) Yüksek boyutlu verilerin olduğu durumlarda, özellikle çok fazla özellik (feature) varsa.

Açıklama: SVM, **yüksek boyutlu veri setlerinde** (özellikle çok fazla özellik olduğunda) **başarılıdır**. SVM, verileri daha yüksek boyutlu bir uzaya taşıyarak **doğrusal olmayan** sınırları doğrusal hale getirebilir ve bu tür durumlarla iyi başa çıkabilir.

---

6. Aşağıdaki ifadelerden hangisi SVM'nin dezavantajlarından biridir?

A) SVM büyük veri setlerinde hızla çalışır.
B) SVM, kernel fonksiyonları sayesinde doğrusal olmayan verilerle de çalışabilir.
C) SVM, büyük veri setlerinde yüksek hesaplama gücü gerektirir.
D) SVM, düşük boyutlu verilerle çalışırken kötü sonuçlar verir.

Doğru Cevap:
C) SVM, büyük veri setlerinde yüksek hesaplama gücü gerektirir.

Açıklama: SVM, büyük veri kümelerinde **eğitim süresi** açısından yavaş olabilir ve hesaplama gücü açısından yoğun kaynak kullanabilir. Bu, büyük veri setlerinde **performans sorunları** yaratabilir.

---

7. Aşağıdaki kernel türlerinden hangisi genellikle doğrusal olmayan verilerle çalışırken kullanılır?

A) Doğrusal kernel
B) Polinomsal kernel
C) RBF (Radial Basis Function) kernel
D) Sigmoid kernel

Doğru Cevap:
C) RBF (Radial Basis Function) kernel

Açıklama: RBF kernel, doğrusal olmayan verilerle çalışırken en yaygın kullanılan kernel türüdür. Bu kernel, verileri daha yüksek boyutlu bir uzaya dönüştürerek doğrusal hale getirir ve böylece doğrusal olmayan sınıflandırmalar yapılabilir.

K-En Yakın Komşu (K-Nearest Neighbors - KNN) algoritması, makine öğrenmesinde basit ve etkili bir sınıflandırma algoritmasıdır. KNN, genellikle hem **sınıflandırma** hem de **regresyon** problemleri için kullanılır. Bu algoritma, yeni bir veri noktası verildiğinde, bu noktayı etrafındaki en yakın **K** komşusuyla karşılaştırarak hangi sınıfa ait olduğunu belirler. Şimdi, **K-En Yakın Komşu** (KNN) algoritmasının temel kavramlarını açıklayarak devam edelim.

K-En Yakın Komşu (KNN) Algoritması

Temel Prensip:

KNN algoritması, **denetimli öğrenme** türünde bir algoritmadır ve aşağıdaki şekilde çalışır:

1. **Yeni veri noktası** (sınıflandırılmak veya tahmin edilmek istenen veri) verildiğinde, bu noktayı eğitim veri kümesindeki diğer noktalarla karşılaştırır.
2. **K** en yakın komşusunu (veri noktasını) belirler. K genellikle pozitif bir tam sayıdır ve kullanıcı tarafından seçilir.
3. **En yakın komşuların çoğunluğu** hangi sınıfta yer alıyorsa, yeni veri noktası o sınıfa atanır (sınıflandırma).
4. Eğer regresyon problemi ise, komşuların **ortalama** veya **ağırlıklı ortalama** değeri alınarak tahmin yapılır.

KNN Algoritmasının Çalışma Prensibi:

1. Mesafe Hesaplama:

KNN algoritması, komşuları seçerken genellikle **mesafe ölçütleri** kullanır. En yaygın kullanılan mesafe ölçütleri şunlardır:

- **Öklidyen Mesafe (Euclidean Distance):**
  - İki nokta arasındaki düz mesafeyi hesaplar.
  - En yaygın kullanılan mesafe türüdür.
  - Formula:

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

- Burada, xxx ve yyy iki veri noktasını ve nnn ise veri noktalarının özellik sayısını ifade eder.

- **Manhattan Mesafesi (Manhattan Distance):**
  - İki nokta arasındaki mesafeyi, sadece yatay ve dikey hareketlerle ölçer.
  - Formula:

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

- **Minkowski Mesafesi:**
  - Daha genel bir mesafe ölçüsüdür ve Öklidyen mesafeye genelleştirilebilir.

2. K Değeri:

- **K**, en yakın kaç komşunun dikkate alınacağını belirler. K'nı yüksek seçmek, modelin daha fazla genelleme yapmasını sağlar, ancak çok küçük K değerleri, modelin gürültüye duyarlı olmasına neden olabilir.
- K'nın seçimi önemlidir. Eğer **K** çok küçükse, model **overfitting** yapabilir, yani eğitilmiş veri kümesindeki özel örnekleri ezberleyebilir. Eğer **K** çok büyükse, model **underfitting** yapabilir, yani verinin karmaşıklığını yeterince öğrenemez.

3. Ağırlıklı Komşular (Opsiyonel):

Bazen, K komşuları **eşit ağırlıklı** değil, mesafe ile **ağırlıklı** olarak kabul edilir. Yani daha yakın komşular daha fazla ağırlık alır.

- **Mesafe Ağırlıklı KNN**: Her komşu, mesafesi ile ters orantılı bir ağırlık alır. Yani daha yakın komşuların katkısı daha fazla olur.

KNN'nin Avantajları ve Dezavantajları

Avantajları:

1. **Basit ve Anlaşılır:** KNN oldukça basit bir algoritmadır, genellikle matematiksel modellemeye gerek duymadan, verilerle doğrudan çalışır.
2. **Esneklik:** Hem **sınıflandırma** hem de **regresyon** problemleri için uygundur.

3. **Eğitim Aşaması Yok:** KNN, **eğitim aşaması** gerektirmez, sadece veri kümesi ile çalışarak hemen tahmin yapabilir.

Dezavantajları:

1. **Hesaplama Maliyeti:** Eğitim aşamasında belirli bir işlem yapılmasa da, tahmin yaparken **uzun süre alabilir**, özellikle büyük veri kümelerinde. Çünkü her yeni veri noktası için tüm eğitim verisiyle mesafe hesaplanır.
2. **Bellek Kullanımı:** KNN, tüm eğitim verisini saklar, bu da büyük veri setlerinde **bellek kullanımı** sorunlarına yol açabilir.
3. **Özellik Ölçekleme:** KNN mesafe ölçtüğü için, özelliklerin ölçeklenmesi önemlidir. Eğer özellikler farklı ölçeklerdeyse, bazı özellikler diğerlerine göre daha baskın olabilir.
4. **Yüksek Boyutluluk Problemi (Curse of Dimensionality):** Özellik sayısının arttığı durumlarda (çok boyutlu veriler), KNN'nin performansı düşebilir.

KNN'nin Kullanım Alanları

- **Sağlık alanı:** Hastalık teşhisi (örneğin, kanserli hücrelerin sınıflandırılması).
- **E-ticaret:** Kullanıcıların geçmiş alışverişlerine dayalı ürün önerileri.
- **Pazar araştırması:** Müşteri segmentasyonu ve sınıflandırması.
- **Yüz tanıma ve görüntü sınıflandırma:** Görüntüler arasındaki benzerliklere dayalı tanıma.

KNN ile İlgili Önemli Parametreler:

1. **K değeri (Komşu Sayısı):** K'nın doğru seçilmesi, modelin başarısı için çok önemlidir.
2. **Mesafe Ölçütü:** Mesafe ölçütü, komşuların nasıl seçileceğini belirler. Öklidyen mesafe çoğunlukla tercih edilir.
3. **Ağırlıklı Komşular:** Daha yakın komşuların daha fazla etkisi olmasını sağlamak.

K-En Yakın Komşu (KNN) algoritmasının temel çalışma prensibi nedir?

A) Verilerin etiketlerini öğrenerek eğitim aşamasında kararlar verir.
B) Veriyi, en yakın K komşusuna bakarak sınıflandırır veya tahmin eder.
C) Modeli eğitim verisiyle eğitir, ancak tahminlerde kullanmaz.
D) Sadece doğrusal verilerle çalışabilir.

Doğru Cevap:
B) Veriyi, en yakın K komşusuna bakarak sınıflandırır veya tahmin eder.

Açıklama: KNN, yeni bir veri noktası verildiğinde, bu noktayı etrafındaki en yakın **K** komşusuyla karşılaştırarak hangi sınıfa ait olduğunu belirler. Yani, sınıflandırma veya regresyon problemleri için **K** komşusunun çoğunluğu dikkate alınır.

---

2. KNN algoritmasında hangi mesafe ölçütü, iki nokta arasındaki en kısa doğrusal mesafeyi ölçer?

A) Manhattan Mesafesi
B) Öklidyen Mesafesi
C) Minkowski Mesafesi
D) Cosine Similarity

Doğru Cevap:
B) Öklidyen Mesafesi

Açıklama: Öklidyen Mesafesi, iki nokta arasındaki doğrusal mesafeyi ölçer ve genellikle en yaygın kullanılan mesafe ölçüsüdür. Matematiksel olarak, iki nokta arasındaki mesafeyi karekök formülüyle hesaplar.

---

3. KNN algoritmasında K değerinin çok büyük seçilmesi neye yol açar?

A) Overfitting (Aşırı Uyum)
B) Underfitting (Yetersiz Uyum)
C) Daha hızlı sonuçlar
D) Eğitim süresinin kısalmasına

Doğru Cevap:
B) Underfitting (Yetersiz Uyum)

Açıklama: Eğer **K** çok büyük seçilirse, model daha fazla komşuyu dikkate alarak genelleştirme yapar. Bu durumda model, veri kümesindeki özel örnekleri göz ardı eder ve

underfitting (yetersiz uyum) meydana gelir. Bu, modelin karmaşık veri örüntülerini öğrenememesi anlamına gelir.

---

4. Aşağıdaki durumların hangisinde KNN algoritması daha verimli olabilir?

A) Çok büyük, etiketlenmiş veri kümeleriyle çalışıldığında.
B) Özelliklerin ölçekleri çok farklı olduğunda.
C) Yüksek doğruluk gerektiren sınıflandırma problemleri.
D) Yeni veriler eklendikçe eğitimin yeniden yapılması gerektiğinde.

Doğru Cevap:
D) Yeni veriler eklendikçe eğitimin yeniden yapılması gerektiğinde.

Açıklama: KNN algoritması, eğitim aşamasında model oluşturmaz. Eğitim verisi her zaman saklanır ve yeni verilerle karşılaşıldığında hemen tahmin yapılabilir. Bu, **yeni veriler eklendikçe yeniden eğitilmesine gerek olmaması** avantajını sağlar. Yani **eğitim süreci sıfırdan yapılmaz**.

---

5. KNN algoritmasının ağırlıklı komşular kullanımının avantajı nedir?

A) Daha uzaktaki komşuların etkisi daha fazla olur.
B) En yakın komşular daha fazla ağırlık alır, bu da daha doğru sonuçlar sağlar.
C) K'nin seçiminden bağımsızdır.
D) Komşuların eşit ağırlıkta olması sağlanır.

Doğru Cevap:
B) En yakın komşular daha fazla ağırlık alır, bu da daha doğru sonuçlar sağlar.

Açıklama: Ağırlıklı komşular, KNN algoritmasında mesafe ile ters orantılı olarak komşulara ağırlık verir. Daha yakın komşulara daha fazla ağırlık verilerek, **daha doğru sınıflandırma** veya tahmin yapılması sağlanır. Bu, özellikle komşular arasında büyük mesafe farkları olduğunda önemlidir.

Karar Ağaçları ve Rastgele Orman Algoritmaları (Decision Trees & Random Forests), makine öğrenmesinde sıklıkla kullanılan güçlü algoritmalardır. Her iki algoritma da sınıflandırma ve regresyon problemlerinde yaygın olarak kullanılır. Şimdi, her iki algoritmayı da detaylıca inceleyelim.

---

1. Karar Ağaçları (Decision Trees)

Temel Prensip:

Karar ağaçları, veriyi **dallanarak** sınıflandırmak veya tahmin yapmak için kullanılan ağaç yapısına dayalı bir algoritmadır. Bu yapıda:

- **Kök düğüm (Root node):** İlk kararın verildiği, en üst düzeydeki düğümdür.
- **Dal (Branch):** Düğümler arasındaki bağlantıdır ve farklı kararları temsil eder.
- **Yaprak düğüm (Leaf node):** Sonuçların bulunduğu düğümlerdir; sınıflandırma için sınıf etiketleri, regresyon için tahmin edilen değerler bulunur.

Karar Ağaçlarının Çalışma Prensibi:

- Veriler, her düğümde belirli bir **özelliğe (feature)** göre ikiye ayrılır.
- Ağaç, bir özellik değerine göre veriyi bölerek her dalda bir karar alır. Bu kararlar, veriyi en iyi şekilde ayıran ve en homojen grupları oluşturan özelliklere dayanır.
- Ağaç, **en iyi bölme** kriterine göre dallanır ve işlem sonunda her yaprakta sınıflandırma veya tahmin yapılır.

Bölme Kriterleri:

- **Gini Index:** Sınıflar arasındaki heterojenliği ölçer. 0'a yakın değerler, homojen gruplara işaret eder.
- **Entropy (Entropi):** Bilgi kazanımı kullanarak veriyi böler. En yüksek bilgi kazanımı ile veriyi böler.
- **Karar Ağacı Düğüm Bölme:** Bir düğümdeki veri noktalarını en iyi şekilde iki sınıfa ayıracak özellik ve sınır değeri belirlenir.

Karar Ağaçlarının Avantajları:

1. **Kolay Anlaşılabilir ve Görselleştirilebilir:** Karar ağaçları, basit görselleştirmelerle kararları net bir şekilde sunar.

2. **Sınıflandırma ve Regresyon için Kullanılabilir:** Hem **sınıflandırma** hem de **regresyon** problemlerine uygulanabilir.
3. **Öznitelik Seçimi:** Ağaçlar, verilerdeki önemli özellikleri otomatik olarak belirler.

Karar Ağaçlarının Dezavantajları:

1. **Overfitting:** Derin ağaçlar, **overfitting** (aşırı uyum) yapabilir, yani eğitim verisine çok iyi uyum sağlar ancak yeni verilere genelleme yapamaz.
2. **Hassasiyet:** Karar ağacı, verilerdeki küçük değişimlere karşı hassas olabilir.
3. **Bölünmüş Veri Kümeleri:** Bazen, veriyi çok küçük parçalara ayırarak anlamlı bir genelleme yapmak zorlaşabilir.

---

2. Rastgele Orman Algoritması (Random Forest)

Temel Prensip:

Rastgele Orman, birden fazla karar ağacının birleşiminden oluşan bir ensemble algoritmadır. Bu algoritma, birçok karar ağacının sonuçlarını birleştirerek daha doğru ve kararlı tahminler yapar. **Bagging (Bootstrap Aggregating)** adı verilen bir yöntemle çalışır:

- Her bir karar ağacı, eğitim verisinin rastgele seçilmiş bir alt kümesiyle eğitilir.
- Her ağaç, veri noktalarını farklı şekilde böler ve kendi kararını verir.
- Sonuçlar, tüm ağaçların oylamasıyla belirlenir. **Sınıflandırma** için çoğunluk oyu, **regresyon** için ortalama tahmin kullanılır.

Rastgele Orman Algoritmasının Çalışma Prensibi:

1. **Bagging:** Her bir karar ağacı, eğitim verisinin farklı bir alt kümesiyle eğitilir.
2. **Özellik Seçimi:** Ağaçlar, her bir düğümde tüm özellikler yerine yalnızca rastgele seçilmiş bir özellik kümesiyle bölünür. Bu, ağaçlar arasındaki çeşitliliği artırır.
3. **Oylama/Ortalama:** Her bir karar ağacının tahmin sonuçları birleştirilir. Sınıflandırma için çoğunluk oyu, regresyon için ortalama kullanılır.

Rastgele Ormanın Avantajları:

1. **Yüksek Genelleme Yeteneği:** Birden fazla ağaç kullanıldığı için, overfitting riski azalır.

2. **Doğal Özellik Seçimi:** Özelliklerin hangilerinin önemli olduğunu belirler ve düşük önem taşıyan özelliklerin etkisini azaltır.
3. **Yüksek Hız ve Doğruluk:** Büyük veri setlerinde bile hızlı ve doğru tahminler yapar.
4. **Hedeflenmiş Parametre Ayarı Gerektirmez:** Rastgele Ormanlar, fazla parametre ayarı gerektirmez ve iyi sonuçlar verir.

Rastgele Ormanın Dezavantajları:

1. **Modelin Yorumlanabilirliği:** Rastgele Orman, karar ağacına göre daha karmaşık bir modeldir ve **yorumlanabilirlik** konusunda daha zordur.
2. **Hesaplama Maliyeti:** Birçok karar ağacı kullanıldığından, hesaplama maliyeti yüksek olabilir. Özellikle büyük veri kümeleriyle çalışırken bu, zorluk çıkarabilir.
3. **Eğitim Süresi:** Karar ağaçlarına göre daha uzun sürede eğitim alabilir.

Karar Ağaçları ve Rastgele Ormanlar ile İlgili Önemli Parametreler:

1. **Karar Ağacı İçin Parametreler:**
    - **Max Depth:** Ağacın derinliği, yani düğüm sayısı.
    - **Min Samples Split:** Bir düğümdeki minimum örnek sayısı.
    - **Max Features:** Her bölme için dikkate alınacak özellik sayısı.
2. **Rastgele Orman İçin Parametreler:**
    - **n_estimators:** Kullanılacak karar ağacı sayısı.
    - **max_features:** Her ağaç için rastgele seçilecek özellik sayısı.
    - **min_samples_split:** Bir düğümde minimum kaç örnek olması gerektiği.

1. Karar Ağaçları algoritmasında hangi kriter kullanılarak en iyi bölme seçilir?

A) Gini Index
B) Euclidean Distance
C) Cross-Validation
D) Feature Importance

Doğru Cevap:
A) Gini Index

Açıklama: Karar ağaçları, her düğümde veriyi bölmek için **Gini Index** veya **Entropy** gibi kriterleri kullanır. Gini Index, sınıflar arasındaki heterojenliği ölçer ve en düşük Gini Index değeri, en iyi bölmeyi ifade eder. Euclidean Distance, KNN algoritmasında kullanılır, Cross-Validation modelin doğruluğunu değerlendirmek için kullanılır, Feature Importance ise modelin önemli özellikleri belirlemesini sağlar.

---

2. Rastgele Orman algoritması, karar ağaçları kullanarak çalışırken hangi yöntemle çalışır?

A) K-Cross Validation
B) Bagging (Bootstrap Aggregating)
C) Boosting
D) Feature Selection

Doğru Cevap:
B) Bagging (Bootstrap Aggregating)

Açıklama: Rastgele Orman algoritması, **bagging (bootstrap aggregating)** yöntemini kullanarak birden fazla karar ağacını oluşturur ve bu ağaçların sonuçlarını oylayarak tahmin yapar. Her ağaç, eğitim verisinin rastgele bir alt kümesiyle eğitilir. Boosting, farklı bir ensemble yöntemidir ve yalnızca karar ağaçlarıyla değil, diğer algoritmalarla da kullanılabilir.

---

3. Karar Ağaçları'nın overfitting yapmasına neden olan faktör nedir?

A) Çok az sayıda özellik kullanılması
B) Çok derin ağaçlar
C) Modelin çok fazla veri üzerinde eğitilmesi
D) Yetersiz eğitim verisi

Doğru Cevap:
B) Çok derin ağaçlar

Açıklama: Overfitting, modelin eğitim verisine çok fazla uyum sağlaması durumudur. Karar ağaçlarında, ağacın derinliği fazla olduğunda, model eğitim verisinin küçük değişikliklerine bile aşırı duyarlı hale gelir. Bu da **overfitting**'e yol açar. Derin ağaçlar, daha fazla bölme yaptığı için eğitim verisinin tüm örüntülerini ezberler.

---

4. Rastgele Orman algoritmasında, her karar ağacı hangi veri kümesi ile eğitilir?

A) Verinin tamamıyla
B) Verinin rasgele seçilmiş bir alt kümesiyle
C) Verinin doğrusal olarak seçilmiş bir alt kümesiyle
D) Verinin zayıf özellikleriyle

Doğru Cevap:
B) Verinin rasgele seçilmiş bir alt kümesiyle

Açıklama: Rastgele Orman algoritmasında her karar ağacı, eğitim verisinin **rasgele seçilmiş bir alt kümesi** ile eğitilir. Bu, bagging (bootstrap aggregating) yönteminin bir parçasıdır. Aynı verilerle eğitilen birden fazla karar ağacının çıktıları birleştirilir. Bu, modelin genelleme yeteneğini artırır.

---

5. Karar Ağaçları, hangi tür problemler için kullanılabilir?

A) Sadece sınıflandırma problemleri
B) Sadece regresyon problemleri
C) Hem sınıflandırma hem regresyon problemleri
D) Sadece lineer problemlere uygulanabilir

Doğru Cevap:
C) Hem sınıflandırma hem regresyon problemleri

Açıklama: Karar ağaçları, **hem sınıflandırma hem regresyon problemlerinde** kullanılabilir. Sınıflandırma için sınıflar arasında ayrım yapar, regresyon için ise sürekli değer tahmin eder.

---

6. Karar Ağaçları algoritmasında entropy (entropi) nedir?

A) Verinin sınıf dağılımındaki belirsizlik ölçüsüdür.
B) Verinin etiketlerine karşılık gelen veri noktalarının toplam sayısıdır.
C) Modelin doğruluğunu ölçen bir parametredir.
D) Ağaçtaki en derin düğüm sayısını ifade eder.

Doğru Cevap:
A) Verinin sınıf dağılımındaki belirsizlik ölçüsüdür.

Açıklama: Entropy, verinin **sınıf dağılımındaki belirsizliği** ölçer. Entropi, bir veri kümesinin ne kadar düzensiz olduğunu belirler. Eğer tüm örnekler aynı sınıfa aitse, entropi 0'dır, yani belirsizlik yoktur. Veri daha homojen oldukça, entropi düşer.

---

7. Rastgele Orman algoritmasında, her karar ağacında hangi özellikler kullanılır?

A) Verinin tüm özellikleri
B) Her bir düğümde, rastgele seçilmiş bir grup özellik
C) Modelin karar verme kriterine göre seçilmiş özellikler
D) Verinin sadece en önemli özellikleri

Doğru Cevap:
B) Her bir düğümde, rastgele seçilmiş bir grup özellik

Açıklama: Rastgele Orman algoritmasında, her karar ağacının her düğümünde, tüm özellikler yerine **rastgele seçilmiş bir grup özellik** kullanılır. Bu, ağaçlar arasında çeşitlilik yaratır ve modelin genel doğruluğunu artırır.

---

8. Karar Ağaçları algoritmasında max_depth parametresi neyi kontrol eder?

A) Modelin doğruluğunu
B) Ağacın ne kadar derinleşebileceğini
C) Her bir düğümde kullanılan özellik sayısını
D) Eğitim verisinin boyutunu

Doğru Cevap:
B) Ağacın ne kadar derinleşebileceğini

Açıklama: max_depth, karar ağacının derinliğini sınırlar. Bu parametre, ağacın en fazla kaç seviyeye kadar dallanabileceğini belirler. Derin ağaçlar daha karmaşık hale gelebilir ve overfitting riskini artırabilir.

---

9. Rastgele Orman algoritmasında, n_estimators parametresi neyi kontrol eder?

A) Her karar ağacındaki düğüm sayısını
B) Kullanılacak karar ağacı sayısını
C) Ağaçların eğitim verisini ne kadar süreyle işleyeceğini
D) Modelin genel doğruluğunu

Doğru Cevap:
B) Kullanılacak karar ağacı sayısını

Açıklama: n_estimators, Rastgele Orman algoritmasında kullanılan karar ağacı sayısını belirler. Daha fazla ağaç, modelin daha kararlı ve doğru sonuçlar üretmesine yardımcı olabilir, ancak aynı zamanda hesaplama maliyeti de artar.

---

10. Karar Ağaçları algoritmasında min_samples_split parametresi neyi kontrol eder?

A) Bir düğümde en az kaç örneğin bulunması gerektiğini
B) Her ağaç için kullanılan özellik sayısını
C) Modelin eğitim süresini
D) Ağacın yaprak sayısını

Doğru Cevap:
A) Bir düğümde en az kaç örneğin bulunması gerektiğini

Açıklama: min_samples_split, karar ağacının bir düğümü bölerken, bölmeyi yapabilmesi için her alt düğümde en az kaç örnek olması gerektiğini belirler. Bu parametre, ağacın aşırı dallanmasını engelleyebilir ve modelin genelleme yeteneğini artırabilir.

---

11. Karar Ağaçları algoritmasında Gini Index nedir?

A) Her düğümdeki örneklerin etiketlerine göre sınıflandırmanın ne kadar karışık olduğunu ölçen bir parametre
B) Modelin doğruluğunu ölçen bir skorlama sistemidir
C) Sadece doğrusal problemlerde kullanılabilen bir ölçüttür
D) Verinin etiketlerine göre eğitim süresini belirleyen bir parametredir

Doğru Cevap:
A) Her düğümdeki örneklerin etiketlerine göre sınıflandırmanın ne kadar karışık olduğunu ölçen bir parametre

Açıklama: Gini Index, bir düğümdeki örneklerin homojenliğini ölçer. Gini değeri 0'a yaklaştıkça, sınıflar arasındaki karışıklık azalır ve düğüm daha homojen hale gelir.