Sınıflandırma

Bir Makine Öğrenimindeki bir sınıflandırıcı, ayrık veya sürekli özellik değerlerinin bir vektörünü giren ve tek bir ayrık değer olan sınıfı çıkaran bir sistemdir. Bir dizi öğenin sınıfını veya kategorisini tahmin etmek için sınıflandırma algoritmaları kullanılır. Sınıflandırma algoritmaları:

Diğer sınıflandırma tekniklerinden bazıları aşağıda verilmiştir:

- K-En Yakın Komşu Algoritması (K-Nearest Neighbour Algorithm)
- Lojistik Regresyon (Logistic Regression)
- Destek Vektör Makineleri (Support Vector Machines)
- Karar Ağaçları (Decision Tree)
- Rasgele Orman Kümeleri (Random Forests)
- Yapay Sinir Ağları (Artifical Neural Networks)
- Naive Bayes

K-En Yakın Komşu Algoritması (K-Nearest Neighbour Algorithm)

K-En Yakın Komşu (KNN), denetimli öğrenme yöntemleri arasında yer alan ve sınıflandırma ile regresyon problemlerinde kullanılan basit ve etkili bir algoritmadır.

1. **Temel Prensip**:
    - KNN, yeni bir veri noktasının sınıfını belirlemek için, eğitim veri setindeki en yakın K komşusuna bakar. Yeni nokta, bu K komşunun çoğunluğuna göre sınıflandırılır.
2. **Adımlar**:
    - **K Değerinin Seçimi**: K değeri belirlenir (örneğin, K = 3).
    - **Mesafe Hesaplama**: Yeni veri noktası ile tüm eğitim veri noktaları arasındaki mesafeler hesaplanır (genellikle Öklidyen mesafe kullanılır).
    - **Komşuların Seçimi**: En yakın K komşu seçilir.
    - **Sınıflandırma veya Tahmin**: K komşunun çoğunluk sınıfı yeni veri noktasının sınıfı olarak atanır.
3. **Avantajları**:
    - Basit ve kolay anlaşılır.
    - Parametrik olmayan bir yöntemdir (veri dağılımı hakkında varsayım yapmaz).
4. **Dezavantajları**:
    - Büyük veri setlerinde yavaş olabilir.
    - K ve mesafe metriği seçimi sonuçları etkiler.

Lojistik Regresyon (Logistic Regression)

Lojistik Regresyon, ikili sınıflandırma problemlerinde yaygın olarak kullanılan bir denetimli öğrenme algoritmasıdır.

1. **Temel Prensip**:
    - Lojistik regresyon, bir olayın olasılığını tahmin etmek için sigmoid fonksiyonu kullanır. Çıktı, 0 ile 1 arasında bir değer alır ve belirli bir eşik değere göre sınıflandırma yapılır.
2. **Matematiksel Model**:
    - Model, $P(y=1|X)=11+e−(\beta 0+\beta 1X1+\beta 2X2+...+\beta nXn)P(y=1|X)=1+e−(\beta 0+\beta 1X1+\beta 2X2+...+\beta nXn)1$ formülüne dayanır.
    - Burada, $\beta 0\beta 0$ sabit terim, $\beta i\beta i$ ise bağımsız değişkenlerin katsayılarıdır.
3. **Avantajları**:
    - Anlaşılması ve uygulanması kolaydır.
    - Olasılık tahminleri sunar.
4. **Dezavantajları**:
    - Lineer olmayan ilişkileri iyi modelleyemez.
    - Sınıflar arasında net ayrımlar olmadığında performansı düşebilir.

Destek Vektör Makineleri (Support Vector Machines)

Destek Vektör Makineleri (SVM), veriyi sınıflandırmak için en uygun ayırıcı hiperdüzlemi bulan güçlü bir denetimli öğrenme algoritmasıdır.

1. **Temel Prensip**:
    - SVM, sınıflar arasındaki marjini maksimize eden bir hiperdüzlem bulur. Bu hiperdüzlem, veri noktalarını sınıflar arasında en iyi şekilde ayırır.
2. **Çekirdek Fonksiyonları**:
    - Lineer ayırıcı hiperdüzlem yeterli olmadığında, çekirdek fonksiyonları (kernel) kullanılarak veri uzayı daha yüksek boyutlara dönüştürülür.
    - Yaygın çekirdekler: Lineer, Polinomial, RBF (Radial Basis Function).
3. **Avantajları**:
    - Karmaşık sınıflandırma problemlerinde etkilidir.
    - Yüksek boyutlu veri setlerinde iyi performans gösterir.
4. **Dezavantajları**:
    - Büyük veri setlerinde hesaplama maliyeti yüksektir.
    - Parametre ayarları ve çekirdek seçimi karmaşık olabilir.

Karar Ağaçları (Decision Trees)

Karar Ağaçları, veriyi dallanma yapıları kullanarak sınıflandıran veya tahmin eden denetimli öğrenme algoritmalarıdır.

1. **Temel Prensip**:
    - Karar ağacı, veri setini özelliklere dayalı olarak bölerek bir ağaç yapısı oluşturur. Her düğüm bir özelliği, her dal bir özelliğin değerini ve her yaprak bir sınıf veya tahmini temsil eder.
2. **Adımlar**:
    - **Bölünme Kriteri**: Hangi özellik üzerine bölüneceğine karar verilir (örneğin, Gini impurity, bilgi kazancı).
    - **Ağaç Oluşturma**: Veri seti, belirli bir kriterle sürekli olarak bölünür.
    - **Ağacı Budama**: Aşırı uyumu önlemek için gereksiz dallar budanır.
3. **Avantajları**:
    - Görselleştirilebilir ve yorumlanabilir.
    - Hem kategorik hem de sayısal verilerle çalışabilir.
4. **Dezavantajları**:
    - Aşırı uyum riski yüksektir.
    - Karar ağacının derinliği büyük veri setlerinde çok fazla olabilir.

Rasgele Orman Kümeleri (Random Forests)

Rasgele Orman (Random Forest), birden fazla karar ağacının oluşturduğu topluluk (ensemble) öğrenme yöntemidir.

1. **Temel Prensip**:
    - Rasgele orman, farklı alt örnekler ve özellik setleri kullanarak birçok karar ağacı oluşturur ve bu ağaçların çoğunluk oyuna veya ortalamasına göre nihai tahmini yapar.
2. **Adımlar**:
    - **Bootstrap Örnekleme**: Eğitim veri setinden rastgele alt örnekler alınır.
    - **Ağaç Oluşturma**: Her alt örnekten bağımsız karar ağaçları oluşturulur.
    - **Tahmin Birleştirme**: Ağaçların tahminleri çoğunluk oyu veya ortalama ile birleştirilir.
3. **Avantajları**:
    - Aşırı uyuma karşı dirençlidir.
    - Yüksek doğruluk sağlar.
4. **Dezavantajları**:
    - Hesaplama maliyeti yüksektir.
    - Yorumlanabilirliği düşüktür.

Yapay Sinir Ağları (Artificial Neural Networks)

Yapay Sinir Ağları (ANN), biyolojik sinir ağlarından esinlenerek geliştirilmiş, özellikle karmaşık desen tanıma ve tahmin problemlerinde kullanılan güçlü algoritmalardır.

1. **Temel Prensip**:
    - ANN, katmanlar halinde düzenlenmiş yapay nöronlardan oluşur. Her nöron, girdi verilerini alır, ağırlıklarla çarpar ve bir aktivasyon fonksiyonu kullanarak çıktı üretir.
2. **Katmanlar**:
    - **Girdi Katmanı**: Giriş verilerini alır.
    - **Gizli Katmanlar**: Veriyi işleyerek öğrenmeyi sağlar.
    - **Çıktı Katmanı**: Nihai tahmini üretir.
3. **Öğrenme**:
    - **Geri Yayılım (Backpropagation)**: Hata sinyalleri kullanılarak ağırlıkların güncellenmesini sağlar.
    - **Aktivasyon Fonksiyonları**: Sigmoid, ReLU, Tanh gibi fonksiyonlar kullanılır.
4. **Avantajları**:
    - Karmaşık ve büyük veri setlerinde etkilidir.
    - Öznitelik mühendisliği gereksinimini azaltır.
5. **Dezavantajları**:
    - Hesaplama maliyeti ve eğitim süresi yüksektir.
    - Büyük miktarda veri gerektirir ve aşırı uyuma meyillidir.

Naive Bayes

Naive Bayes, Bayes teoremini temel alarak çalışan, özelliklerin birbirinden bağımsız olduğunu varsayan bir denetimli öğrenme algoritmasıdır.

1. **Temel Prensip**:
    - Naive Bayes, her özelliğin sınıfa bağımsız katkıda bulunduğunu varsayar ve bu bağımsızlık varsayımı altında olasılık hesaplamaları yapar.
2. **Bayes Teoremi**:
    - $P(A|B)=P(B|A)\cdot P(A)P(B)P(A|B)=P(B)P(B|A)\cdot P(A)$
3. **Naive Varsayım**:
    - Özellikler birbirinden bağımsızdır (bu varsayım genellikle gerçekçi olmasa da pratikte iyi sonuçlar verir).
4. **Adımlar**:
    - Her sınıf için öncelik olasılıkları hesaplanır.
    - Her özellik için sınıf koşullu olasılıkları hesaplanır.
    - Yeni bir veri noktası için bu olasılıklar birleştirilir ve en yüksek olasılığa sahip sınıf seçilir.
5. **Avantajları**:
    - Hızlı ve verimlidir.
    - Metin sınıflandırma gibi problemlerde etkilidir.
6. **Dezavantajları**:
    - Özelliklerin bağımsızlığı varsayımı gerçekçi değildir.
    - Sürekli verilerde performans düşebilir.

Bu algoritmalar, makine öğrenmesinin çeşitli problemlerinde kullanılmak üzere geliştirilmiş olup, her birinin kendi avantajları ve sınırlamaları vardır. Hangi algoritmanın kullanılacağı, verinin yapısına ve çözülmesi gereken problemin özelliklerine bağlıdır.