Sınıflandırma Karar Ağaçları :

Sınıflandırma:

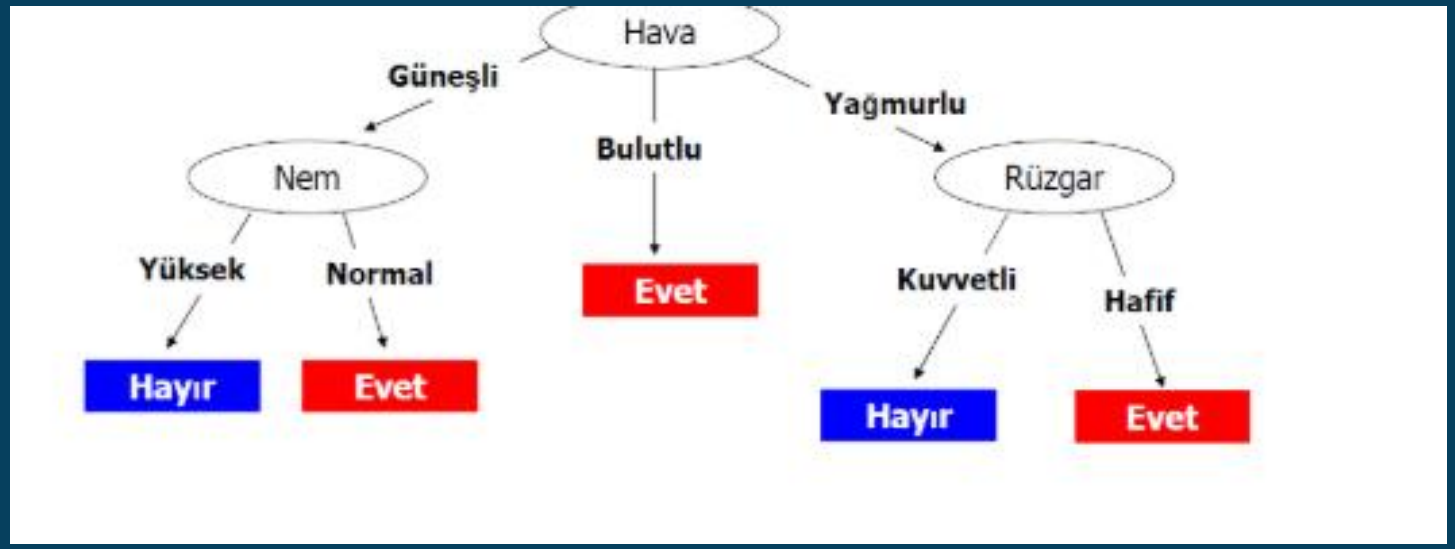
Bir Makine Öğrenimindeki bir sınıflandırıcı, ayrık veya sürekli özellik değerlerinin bir vektörünü giren ve tek bir ayrık değer olan sınıfı çıkaran bir sistemdir. Bir dizi öğenin sınıfını veya kategorisini tahmin etmek için sınıflandırma algoritmaları kullanılır. Sınıflandırma algoritmaları: Diğer sınıflandırma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

Karar Ağaçları :

- Karar ağacı algoritması, denetimli öğrenme kategorisine girer ve hem regresyon hem de sınıflandırma problemlerini çözmek için kullanılabilir. Bu algoritma, her yaprak düğümün bir sınıf etiketine karşılık geldiği ve özniteliklerin ağacın iç düğümünde temsil edildiği sorunu çözmek için ağaç temsilini kullanır. Ayrıca, herhangi bir boole fonksiyonunu ayrık öznitelikler üzerinde temsil etmek için de kullanılabilir.
- Karar ağacı öğrenmesi, bir öğenin hedef değeri ile ilgili sonuçlara gitmek için bir tahmin modeli olarak kullanılır. Hedef değişkenin ayrı bir değer kümesi alabileceği ağaç modellerine sınıflandırma ağaçları denir; bu ağaç yapılarında yapraklar sınıf etiketlerini ve dallar bu sınıf etiketlerine yol açan özelliklerin birleşimlerini temsil eder. Hedef değişkenin sürekli değerler alabileceği karar ağaçlarına (tipik olarak gerçek sayılar) regresyon ağaçları denir.
- Karar ağaçları, giriş verisinin bir algoritma yardımıyla gruplara bölünerek tüm elemanlarının aynı sınıf etiketine sahip olması için yapılan sınıflama işlemidir. Bu, giriş verisinin bir kümeleme algoritması yardımıyla tekrar tekrar gruplara bölünmesine dayanır. Grubun tüm elemanları aynı sınıf etiketine sahip olana kadar kümeleme işlemi derinlemesine devam eder. Karar ağaçları, kararları ve karar alma süreçlerini görsel ve açık bir şekilde temsil etmek için kullanılabilir ve veri madenciliğinde verilerin tanımlanması ve sonuçların sınıflandırılması için de kullanılabilir.

- Karar ağaçları metotu, giriş verisinin bir algoritma yardımıyla gruplara bölünerek tüm elemanlarının aynı sınıf etiketine sahip olması için yapılan sınıflama işlemidir. Giriş verisinin bir kümeleme algoritması yardımıyla tekrar tekrar gruplara bölünmesine dayanır. Grubun tüm elemanları aynı sınıf etiketine sahip olana kadar kümeleme işlemi derinlemesine devam eder.



Karar ağacı kullanılırken yapılan bazı varsayımlar aşağıdadır:

- Başlangıçta, tüm eğitim seti kök olarak kabul edilir.
- Özellik değerlerinin kategorik olması tercih edilir. Değerler sürekli ise, model oluşturmadan önce ayrıklaştırılırlar.
- Öznitelik değerleri temelinde, kayıtlar özyinelemeli olarak dağıtılır.
- Öznitelikleri kök veya dahili düğüm olarak sıralamak için istatistiksel yöntemler kullanılır**.**

Karar ağacı tipleri ikiye ayrılır:

Entropiye dayalı sınıflandırma ağaçları (ID3, C4.5)

Regresyon ağaçları (CART).

Karar Ağacı Algoritması:

- Karar ağaçları eğiticili öğrenme için çok yaygın bir yöntemdir. Algoritmanın adımları:

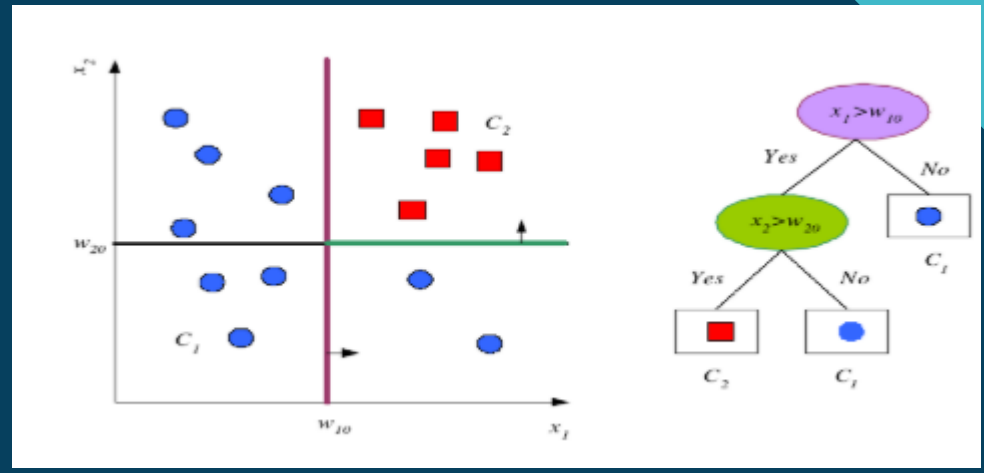
1) T öğrenme kümesini oluşturulur.
2) T kümesindeki örnekleri en iyi ayıran nitelikler belirlenir.
3) Seçilen nitelik ile ağacın düğümleri oluşturulur ve herbir düğümde alt düğümler veya ağacın yapraklarını oluşturulur. Alt düğümlere ait alt veri kümesinin örneklerini belirlenir
4) 3. adımda oluşturulan her alt veri kümesi için

   Örneklerin hepsi aynı sınıfa aitse

   Örnekleri bölecek nitelik kalmamışsa

   Kalan niteliklerin değerini taşıyan örnek yoksa işlemi sonlandır.

   Diğer durumda alt veri kümesini ayırmak için 2. adımdan devam edilir.



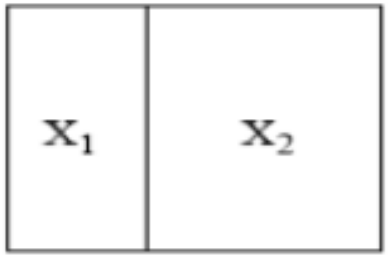
Ezber (Overfitting: Aşırı Uyum):

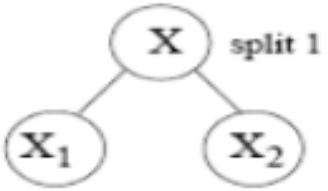
- Tüm makine öğrenmesi yöntemlerinde verinin ana hatlarının modellenmesi esas alındığı için öğrenme modelinde ezberden (overfitting) kaçınılmalıdır.
- Tüm karar ağaçları önlem alınmazsa ezber yapar. Bu yüzden ağaç olusturulurken veya olusturulduktan sonra budama yapılmalıdır.

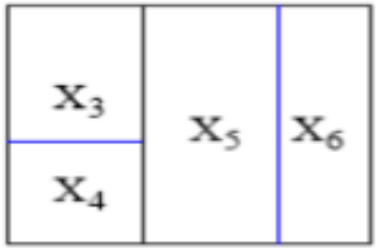
Ağaç budama, karar ağacından sınıflandırmaya katkısı olmayan bölümlerin çıkarılması işlemidir. Bu sayede karar ağacı hem daha sade hem de daha anlaşılabilir hale gelir. İki tür budama yöntemi vardır: ön budama ve sonradan budama.

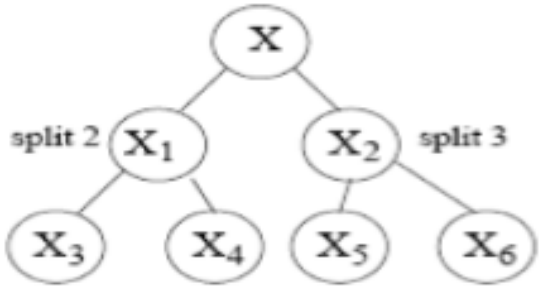
- Ön budama işlemi ağaç oluşturulurken yapılır. Bölünen niteliklerin değerleri belirli bir eşik değerinin (hata toleransının) üstünde değilse, ağaç bölme işlemi durdurulur ve o an elde bulunan kümedeki baskın sınıf etiketi, yaprak olarak oluşturulur.
- Sonradan budama işlemi ise ağaç oluşturulduktan sonra devreye girer. Alt ağaçları silerek yaprak oluşturma, alt ağaçları yükseltme, dal kesme şeklinde yapılabilir. Aşırı uyumu önlemek için ağacı büyütmeyi durdurabiliriz, ancak bu kriter miyop olma eğilimindedir. Bu nedenle standart yaklaşım, "dolu" bir ağaç yetiştirmek ve ardından budama yapmaktır.

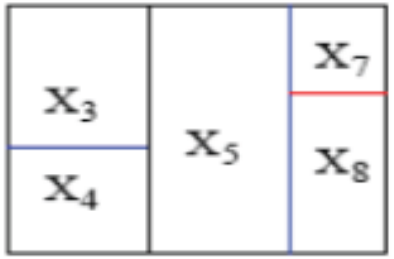
Sınıflandırma Ağaçları:

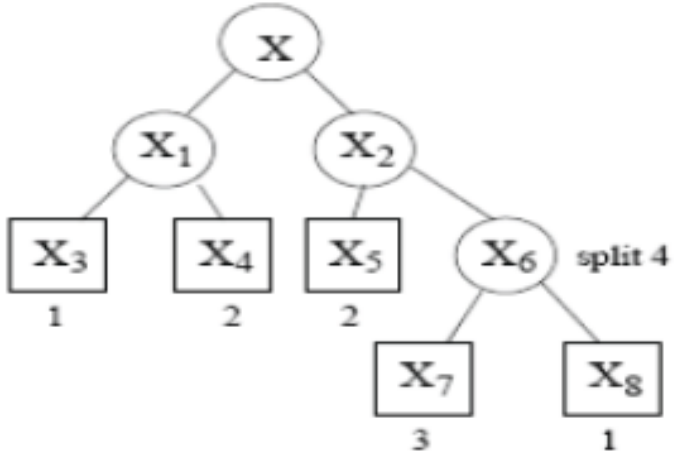


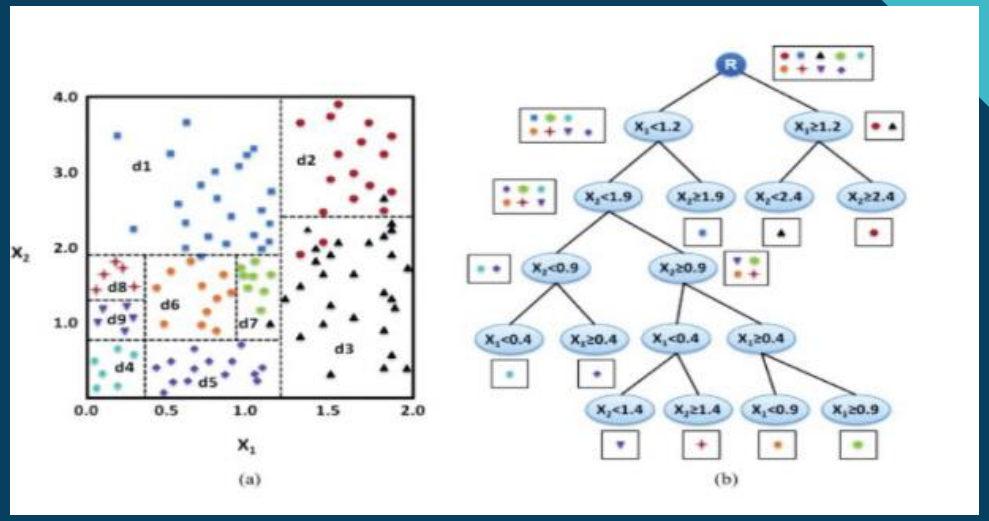












- Karar Ağacında en büyük zorluk, her seviyede kök düğüm için özniteliğin tanımlanmasıdır. Bu işlem öznitelik seçimi olarak bilinir. İki popüler öznitelik seçim ölçüsü bulunmaktadır:
- 1) Bilgi Kazancı
- 2) Gini İndeksi 1

- 1) Bilgi Kazancı Eğitim örneklerini daha küçük alt kümelere bölmek için karar ağacında bir düğüm kullandığımızda entropi değişir. Bilgi kazancı, entropideki bu değişimin bir ölçüsüdür. Tanım: Diyelim ki S bir örnekler kümesi, A bir nitelik, Sv, S'nin A = v ile alt kümesi ve Değerler (A), A'nın tüm olası değerlerinin kümesidir, o zaman

$$Gain(S, A) = Entropy(S) - \sum_{veValues(A)} \frac{|S_v|}{|S|}.Entropy(S_v)$$

Bilgi Kazanımını Kullanarak Karar Ağacı Oluşturma Gereklilikler:

Kök düğümle ilişkili tüm eğitim örnekleriyle başlanır

Her bir düğümün hangi öznitelikle etiketleneceğini seçmek için bilgi kazancı kullanılır.

Not: Hiçbir kökten yaprağa yol, aynı ayrık özniteliği iki kez içermemelidir

Her alt ağacı, ağaçta o yolda sınıflandırılacak eğitim örneklerinin alt kümesinde yinelemeli olarak oluşturulur.

Sınır vakaları:

Tüm pozitif veya tüm negatif eğitim örnekleri kalırsa, o düğümü buna göre “evet” veya “hayır” olarak etiketlenr.

Hiçbir öznitelik kalmazsa, o düğümde kalan eğitim örneklerinin çoğunluk oyu ile etiketlenir. Örnek kalmadıysa, ebeveynin eğitim örnekleri çoğunluk oyu ile etiketlenir.

ID3 Algoritması:

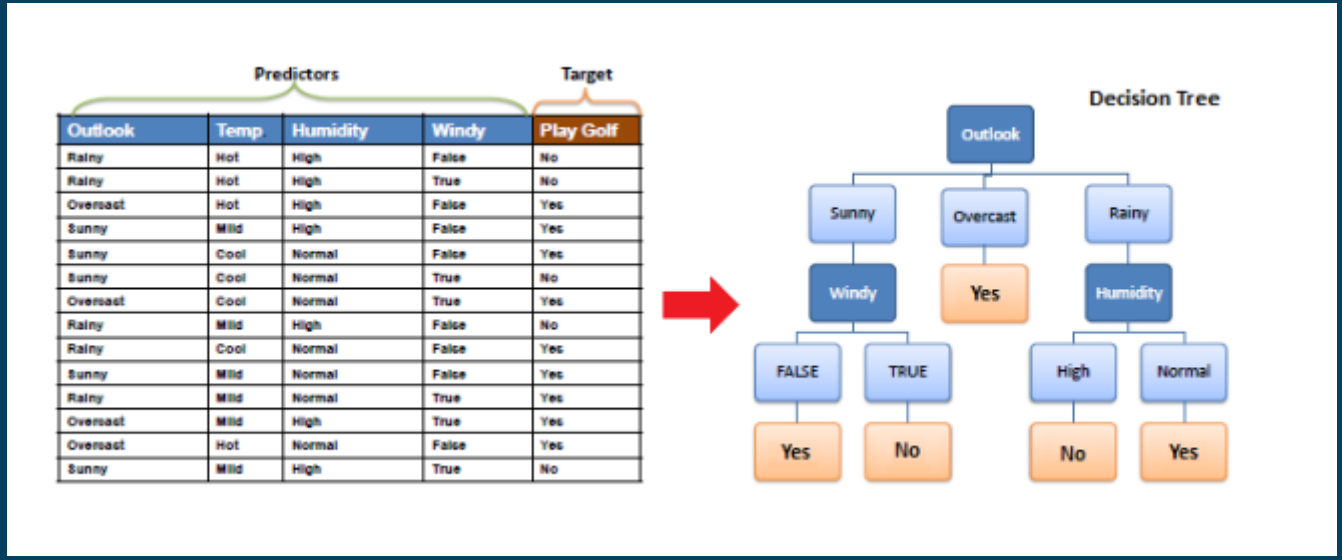
- Sadece kategorik veri ile çalısan bir yöntemdir. Her iterasyonun ilk adımında veri örneklerine ait sınıf bilgilerini tasıyan vektörün entropisi belirlenir. Daha sonra özellik vektörlerinin sınıfa bağımlı entropileri hesaplanarak ilk adımda hesaplanan entropiden çıkartılır. Bu sekilde elde edilen değer ilgili özellik vektörüne ait kazanç değeridir. En büyük kazanca sahip özellik vektörü ağacın o iterasyonda belirlenen dallanmasını gerçeklestirir.

C4.5 Algoritması:

- ID3 algoritmasının nümerik özellik içeren veriye uygulanabilen seklidir. ID3'ten tek farkı nümerik özelliklerin kategorik hale getirilebilmesini sağlayan bir esikleme yöntemini içermesidir. Temel mantık nümerik özellik vektöründeki tüm değerler ikili olarak ele alınarak ortalamaları esik olarak denenir. Hangi esik değeriyle bilgi kazanımı en iyi ise o değer seçilir. Seçilen esiğe göre özellik vektörü kategorize edilir ve ID3 uygulanır.

Example: Decision Tree - Classification :

- Karar ağacı, bir ağaç yapısı şeklinde sınıflandırma veya regresyon modelleri oluşturur. Bir veri kümesini giderek daha küçük alt kümelere ayırırken aynı zamanda ilgili bir karar ağacı aşamalı olarak geliştirilir. Nihai sonuç, karar düğümleri ve yaprak düğümleri olan bir ağaçtır. Bir karar düğümünün iki veya daha fazla şubesi vardır (örn. Güneşli, Bulutlu ve Yağmurlu). Yaprak düğümü bir sınıflandırmayı veya kararı temsil eder. Kök düğüm adı verilen en iyi tahmin ediciye karşılık gelen bir ağaçtaki en üstteki karar düğümü. Karar ağaçları hem kategorik hem de sayısal verileri işleyebilir.



Entropy :

- Bir karar ağacı, bir kök düğümden yukarıdan aşağıya oluşturulur ve verileri benzer değerlere sahip (homojen) örnekler içeren alt kümelere ayırmayı içerir. ID3 algoritması, bir örneğin homojenliğini hesaplamak için entropiyi kullanır. Örnek tamamen homojen (Olma olasılığı 1 ise olmma olasılığı sıfırdır. Tüm olasıklar toplamı bire eşit olmak zorundadır) ise entropi sıfırdır ve örnek eşit olarak bölünmüşse entropisi birdir.