

KARAR AĞACI ALGORİTMASI:

Sınıflandırma, makine öğreniminde iki adımlı bir süreç, öğrenme adımı ve tahmin adımıdır. Öğrenme aşamasında, model verilen eğitim verilerine dayanarak geliştirilmiştir. Tahmin aşamasında, model verilen veriler için yanıtı tahmin etmek için kullanılır. Karar ağacı, anlaşılması ve yorumlanması en kolay ve popüler sınıflandırma algoritmalarından biridir.

Karar ağacı algoritması, Danışmanlı öğrenme algoritmaları ailesine aittir. Diğer danışmanlı öğrenme algoritmalarının aksine, karar ağacı algoritması regresyon ve sınıflandırma problemlerini çözmek için de kullanılabilir. Bir karar ağacı kullanmanın amacı, önceki verilerden (eğitim verileri) çıkarılan basit karar kurallarını öğrenerek hedef değişkenin sınıfını veya değerini tahmin etmek için kullanılabilecek bir eğitim modeli oluşturmaktır.

Karar ağaçlarında, bir kayıt için bir sınıf etiketini tahmin etmek için ağacın kökünden işleme başlanır. Kök özniteliğinin değerleri, kaydın özniteliğiyle karşılaştırılır. Karşılaştırmaya dayanarak, bu değere karşılık gelen dalı takip edilir ve bir sonraki düğüme geçilir.

Karar ağaçlarının türleri, sahip olunan hedef değişkenin türüne dayanmaktadır. İki tür karar ağacı vardır:

- Kategorik değişkenli karar ağacı: kategorik bir hedef değişkene sahip olan karar ağacıdır.
- Sürekli değişkenli karar ağacı: karar ağacının sürekli bir hedef değişkeni olduğu durumdur.

Karar ağaçları ile ilgili önemli terimler:

Kök düğüm: tüm popülasyonu veya örneği temsil eder ve bu daha sonra iki veya daha fazla homojen kümeye bölünür.

Bölme: bir düğümü iki veya daha fazla alt düğüme bölme işlemidir.

Karar düğümü: bir alt düğüm daha fazla alt düğümlere ayrıldığında, buna karar düğümü denir.

Yaprak / Terminal düğümü: Bölünmeyen düğümlere yaprak veya Terminal düğümü denir.

Budama: bir karar düğümünün alt düğümlerini kaldırdığımızda, bu işleme budama denir.

Bölme işleminin tam tersini söyleyebilirsiniz.

UİST468.MAKİNE ÖĞRENMESİ

Şube / alt ağaç: tüm ağacın bir alt bölümüne dal veya alt ağaç denir.

Üst (parent) ve alt (child) düğüm: alt düğümlere bölünmüş bir düğüm, alt düğümlerin üst düğümü olarak adlandırılırken, alt düğümler bir üst düğümün alt düğümüdür.

Akış şemalarına benzeyen karar ağaçlarında sınıflandırma için bir ağaç oluşturulmaktadır. Kök düğüm, iç düğüm, dal ve yaprak düğüm gibi yapılardan oluşan bu ağaca veri tabanındaki her bir gözlem uygun kurallara göre yerleştirilmektedir. Veri tabanındaki her bir özellik bir “düğüm” tarafından temsil edilmektedir. En üstte yer alan düğüm “kök düğüm” olarak isimlendirilmektedir. En son düğüm “yaprak düğüm” olarak isimlendirilirken, aynı zamanda bir sınıfı temsil etmektedir. Kök düğüm ve yaprak düğüm arasındaki düğümler ise “iç düğüm” olarak isimlendirilmektedir. Kök düğüm, iç düğüm ve yaprak düğümler arasında kalan yapılar ise “dal” olarak adlandırılmaktadır. Her düğüm kendinden sonra dal ya da dallara ayrılmaktadır. Bu ayrılma işlemi sürecinde ilgili düğüm hakkında cevabı veri tabanında bulunacak bir soru sorulmakta ve verilen yanıtı göre bir dal izlenmektedir.



Karar ağaçları, örnekleri kökten bazı yaprak/terminal düğümüne kadar ağaçtan ayırarak sınıflandırır. Ağaçtaki her düğüm, bazı öznitelikler için bir test durumu görevi görür ve düğümden inen her kenar, test durumuna Olası yanıtlara karşılık gelir. Bu işlem doğada özyinelemeli ve yeni düğümde kök salmış her alt ağaç için tekrarlanır.

Aşağıda, karar ağacını kullanırken yaptığımız varsayımlardan bazıları bulunmaktadır:

- Başlangıçta, tüm eğitim seti kök olarak kabul edilir.
- Özellik değerleri kategorik olarak tercih edilir. Değerler sürekli ise, modeli oluşturmadan önce ayrıklaştırılırlar.

UİST468.MAKİNE ÖĞRENMESİ

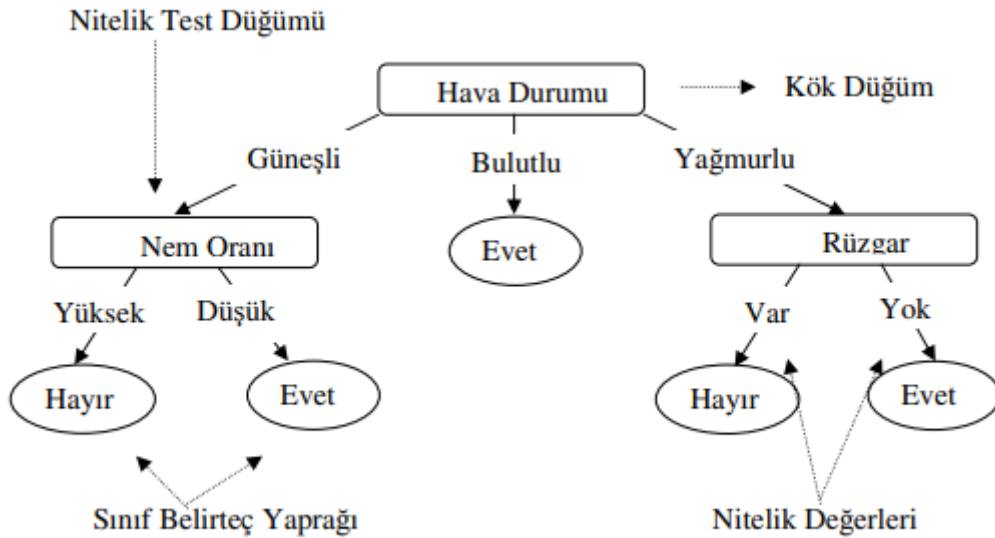
- Kayıtlar, öznitelik değerlerine göre özyinelemeli olarak dağıtılır.
- Öznitelikleri ağacın kök veya iç düğümü olarak yerleştirmek için bazı istatistiksel yaklaşım kullanılarak yapılır.

Karar ağaçlarına ilişkin bazı kavramları bilinen bir veri seti üzerinden açıklayalım. Balık tutmaya gidip gitmeme kararı verirken hava durumu, sıcaklık, nem ve rüzgar durumu gibi meteorolojik verileri göz önüne alınır.

HAVA DURUMU	SICAKLIK	NEM (&)	RÜZGÂR	BALIĞA GİTME
Güneşli	30	80	Var	HAYIR
Güneşli	35	70	Yok	HAYIR
Güneşli	25	55	Var	EVET
Güneşli	32	58	Yok	EVET
Güneşli	35	55	Var	EVET
Bulutlu	25	68	Yok	EVET
Bulutlu	28	74	Yok	EVET
Bulutlu	26	76	Var	EVET
Bulutlu	24	75	Yok	EVET
Bulutlu	21	78	Var	EVET
Yağmurlu	21	90	Var	HAYIR
Yağmurlu	18	85	Var	HAYIR
Yağmurlu	20	87	Yok	EVET
Yağmurlu	21	82	Yok	EVET

Böyle bir karar problemi için aşağıdaki gibi bir karar ağacı kullanılabilir.

UİST468.MAKİNE ÖĞRENMESİ



Karar ağaçları oluşturulurken hangi algoritmanın kullanıldığı çok önemlidir. Kullanılan algoritma çeşidine bağlı olarak oluşturulan ağaç yapısı farklılık göstermektedir. Bundan dolayı farklı ağaç yapıları sınıflandırma işlemini gerçekleştirirken farklı kurallar üretir. Karar ağaçlarına dayalı olarak geliştirilen algoritmalar kök düğümün ve diğer düğümlerin seçimlerinde izledikleri yol açısından birbirlerinden ayrılırlar, ancak bazı noktalarda ise birbirlerine benzemektedirler ya da benzerlik gösterirler.

Karar ağaçları hedef değişkeni tahmin etmede pek çok test gerçekleştirerek en iyi sırayı kullanır. Karar ağaçlarında kullanılan temel algoritma adımları birbirlerine benzemektedir. Başlangıçta kök düğüm (root) oluşturulur. Eğer tüm örnekler aynı sınıfa ait ise düğüm yaprağa dönüşür, değilse test yapılarak bölünme gerçekleştirilir dallar (branch) oluşturulur. Dalları diğer testlerin gerçekleşmesine neden olur ve test işlemi yapraklarda (leaf) sonlandırılır. Örneklerin her bölünmesinde aynı algoritmalar kullanılır. Genellikle kategorik değişkenler kullanılır. Eğer veriler sürekli ise kategorik hale dönüştürülür. Bölümlemenin sonlandırılması aşağıdaki adımlardan biri meydana geldiğinde gerçekleşir;

- Bir düğümde bulunan bütün örnekler aynı sınıfa ait ise,
- Örnekleri bölecek nitelik kalmadığında,
- Başka örneklem kalmadığında.

Karar ağacı oluşturmak için çeşitli algoritmalar geliştirilmiştir. Geliştirilen bu algoritmalar içerisinde CHAID (Chi-Squared Automatic Interaction Detector), CART (Classification and Regression Trees), ID.3, ID.5, MARS (Multivariate Adaptive Regression Splines), QUEST

UİST468.MAKİNE ÖĞRENMESİ

(Quick, Unbiased, Efficient Statistical Tree), C5.0, SLIQ (Supervised Learning in Quest), SPRINT (Scalable Parallelizable Induction of Decision Trees) algoritmaları başlıcalarıdır.

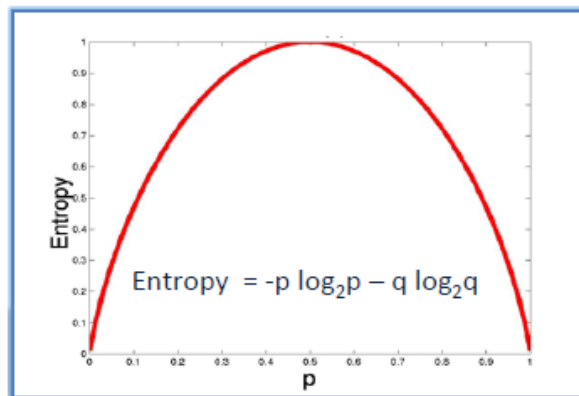
ID3 algoritması:

İlk olarak J. Ross Quinlan tarafından Sydney Üniversitesinde geliştirilen ID3 algoritması, diğer değişkenler arasından, sınıflamada kullanılacak en ayırt edici değişkeni bulurken entropi kavramından yararlanır. Entropi;

- Eldeki bilginin sayısallaştırılması,
- Beklentisizliğin maksimumlaştırılması,
- Belirsizlik ve rastgeleliğin ölçülmesi için kullanılır.

Entropi, 0-1 arasında bir değer alır. Elimizdeki bütün veriler tek bir sınıfa ait olduğunda entropi sıfır olur. Örneğin herkes aynı futbol takımını tuttuğu bir durumda rastgele birine hangi takımı tuttuğunu sorduğumuzda alacağımız cevap belli olacaktır yani entropi sıfır değerini alır. Bütün olasılıklar eşit olduğu zaman entropi maksimum değerine ulaşır. Matematiksel olarak aşağıdaki gibi ifade edilir

$$H(T) = -\sum p_i \log_2(p_i)$$



$$\text{Entropy} = -0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$$

Karar ağacının oluşturulacağı veri setinin tamamının entropisi hesaplanır; ancak dallanma gerçekleştiği zaman bölümlere ayrılan her bir bölüm için entropinin yeniden hesaplanması gerekir. Buradaki $H(T)$ hedef niteliği temsil eder, p_i ise düğüm içinde i sınıfında olma olasılığıdır.

ID3 algoritması, veri tabanında doğru dallandırma yapmak için bilgi kazancı (information gain) kullanır. Veri tabanında dallandırmalar oluşukça doğru sınıflandırmalar için gereken bilgi azalacaktır. Kazanımı hesaplamak için: verilerin ham halinin entropisi ile her bir nitelik için hesaplanan entropilerin ağırlık toplamı arasındaki fark alınır. Bu fark hangi alt bölüm için büyük ise o alt bölüme doğru dallandırma işlemi gerçekleştirilir. Bilgi kazancının formülü aşağıdaki gibi ifade edilir;

$$\text{Kazanç}(X,T) = H(T) - H(X,T) \quad H(X,T) = \sum_{i=1}^n \frac{|T_i|}{|T|} H(T_i)$$

Buradaki amaç Kazanç (X,T) değerini maksimum yapan değeri bulmaktır.

Örnek:

Hava Durumu	Sıcaklık	Nem Oranı	Rüzgar	Balığa gitme
güneşli	sıcak	yüksek	yok	hayır
güneşli	sıcak	yüksek	var	hayır
bulutlu	sıcak	yüksek	yok	evet
yağmurlu	ılık	yüksek	yok	evet
yağmurlu	soğuk	normal	yok	evet
yağmurlu	soğuk	normal	var	hayır
bulutlu	soğuk	normal	var	evet
güneşli	ılık	yüksek	yok	hayır
güneşli	soğuk	normal	yok	evet
yağmurlu	ılık	normal	yok	evet
güneşli	ılık	normal	Var	evet
bulutlu	ılık	yüksek	var	Evet
bulutlu	sıcak	normal	yok	evet
yağmurlu	ılık	yüksek	var	hayır

Veri kümesinde beş nitelik vardır: “Hava durumu”, “sıcaklık”, “nem oranı”, “rüzgar”, “balığa gitme” nitelikleri. Amaç bu verilerden yararlanarak karar ağacı oluşturmak ve daha sonra “balığa gitme” niteliği boş olan kayıtlarda bu niteliği tahmin etmektir. Burada “balığa gitme” çıkış niteliği, diğerleri giriş nitelikleri olarak düşünülebilirler. Aşağıdaki veri kümesindeki nitelikler ve alabileceği değerler gösterilmiştir.

Nitelik	Alabileceği Değerler
Hava Durumu	güneşli, bulutlu, yağmurlu
Sıcaklık	sıcak, ılık, soğuk
Nem Oranı	normal, yüksek
Rüzgar	var, yok
Balığa gitme	evet, hayır

Oyun oyna alanı iki farklı değer içermektedir (evet, hayır). C1 sınıfının “evet” değerini, C2 sınıfının “hayır” değerini temsil ettiği varsayılırsa, veri kümesinde C1 sınıfı için 9, C2 sınıfı için 5 kaydın olduğu görülür. Veri kümesini C1 ve C2 sınıflarına ayırmak için gerekli olan bilgi aşağıdaki şekilde hesaplanır

$$H(s_1, s_2) = I(9,5) = - 9/14 \log_2 9/14 - 5/14 \log_2 5/14 = 0.940$$

Daha sonra verideki her nitelik için entropy değerleri hesaplanır. Hava durumu niteliği için entropy değerinin hesaplanması şu şekildedir: Bu nitelik 3 farklı değer alabilmektedir (güneşli, bulutlu, yağmurlu). Her farklı değer için gerekli bilgi miktarı hesaplanır. Veri kümesinde hava durumunun güneşli olduğu kayıtlarda, iki “evet” ve üç “hayır” değeri bulunmaktadır. Gerekli bilgi miktarının hesaplanması:

$$H(2,3) = - 2/5 \log_2 2/5 - 3/5 \log_2 3/5 = 0.971$$

Veri kümesinde hava durumunun bulutlu olduğu kayıtlarda, sadece dört “evet” değeri bulunmaktadır, “hayır” değeri hiç yoktur. Gerekli bilgi miktarının hesaplanması:

$$H(4,0) = - 4/4 \log_2 4/4 - 0/4 \log_2 0/4 = 0$$

Veri kümesinde hava durumunun yağmurlu olduğu kayıtlarda, üç “evet” ve iki “hayır” değeri bulunmaktadır. Gerekli bilgi miktarının hesaplanması:

$$H(3,2) = - \frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} = 0.971$$

Üç farklı durum için bilgi miktarları hesaplandıktan sonra entropy değeri hesaplanır:

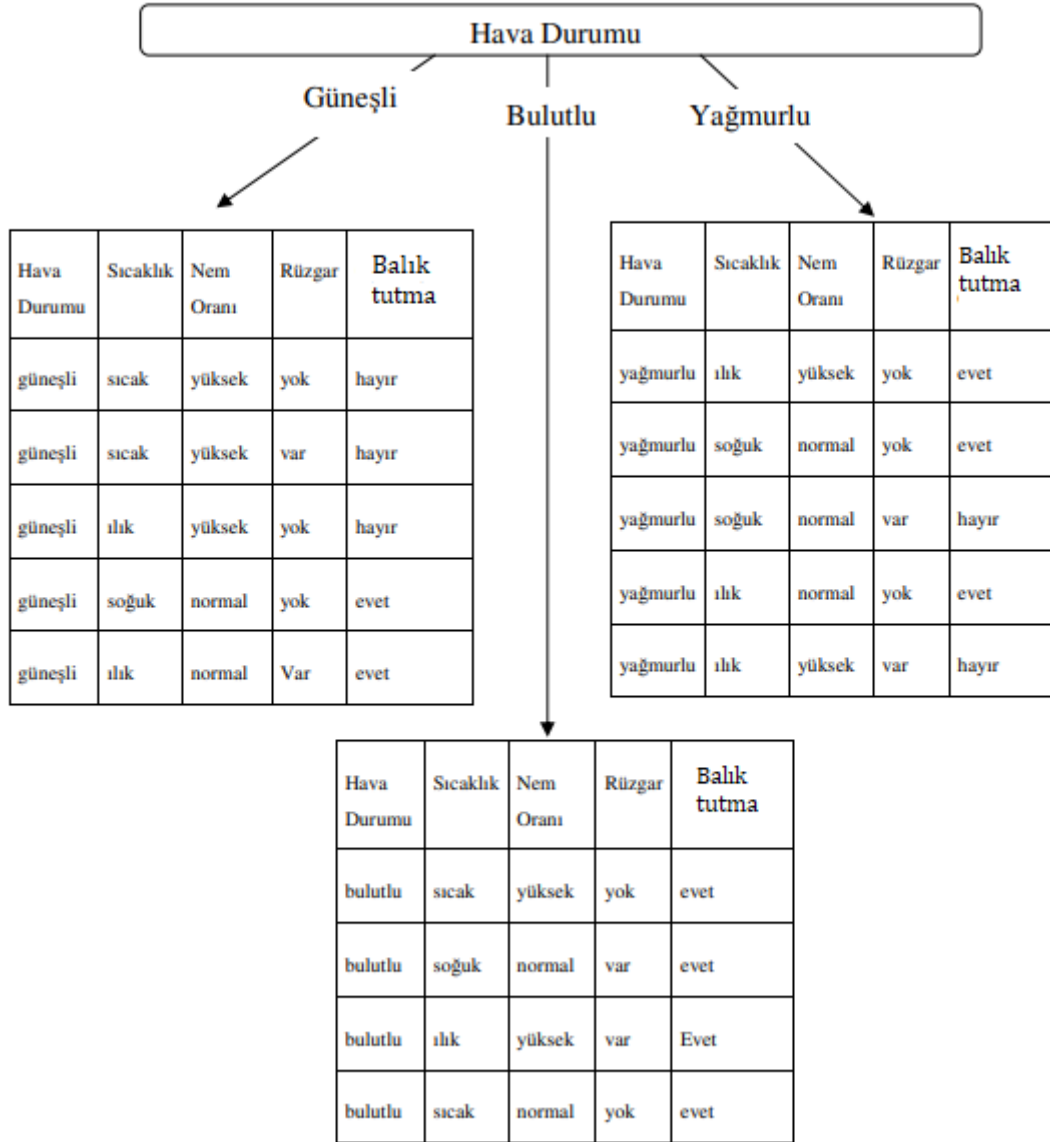
$$E(\text{hava durumu}) = \frac{5}{14} H(2,3) + \frac{4}{14} H(4,0) + \frac{5}{14} H(3,2) = 0.694$$

Karar ağacında hava durumu niteliği kullanılarak bir sınıflama yapıldığında elde edilecek bilgi kazancı şu şekilde hesaplanır:

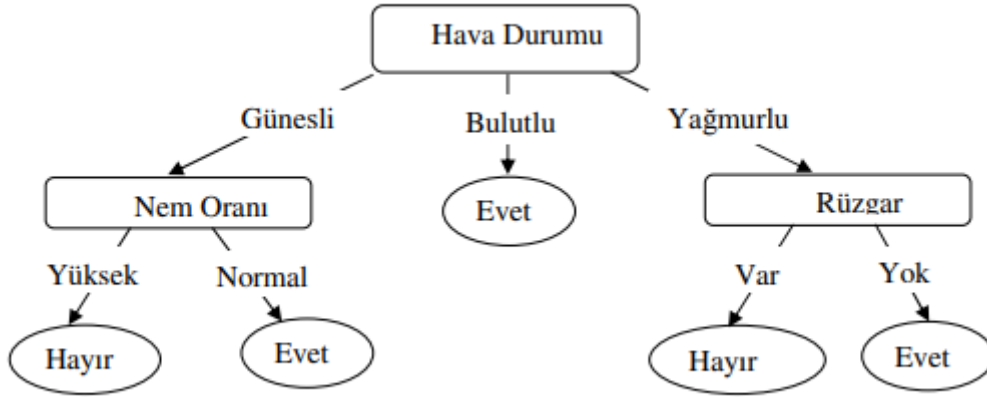
Gain (hava durumu) = $H(s_1, s_2) - E(\text{hava durumu}) = 0.940 - 0.694 = 0.246$. Benzer şekilde diğer kazançlar aşağıdaki gibi elde edilir.

Bilgi Kazancı	Değerler
Gain(havadurumu)	0.246
Gain(sıcaklık)	0.029
Gain(nemoranı)	0.151
Gain(rüzgar)	0.151

Hava durumu niteliğine ait bilgi kazancı, hesaplanan bilgi kazançları arasında en yüksek değere sahip olduğu için karar düğümü olarak seçilir. Düğümün adına niteliğin adı verilir ve sahip olduğu değerler için dallar oluşturulur. “Hava durumu” niteliğinin alabileceği farklı değerler “güneşli”, “bulutlu” ve “yağmurlu” olduğu için veri, aşağıda görüldüğü gibi üç alt veri kümesine ayrılmıştır



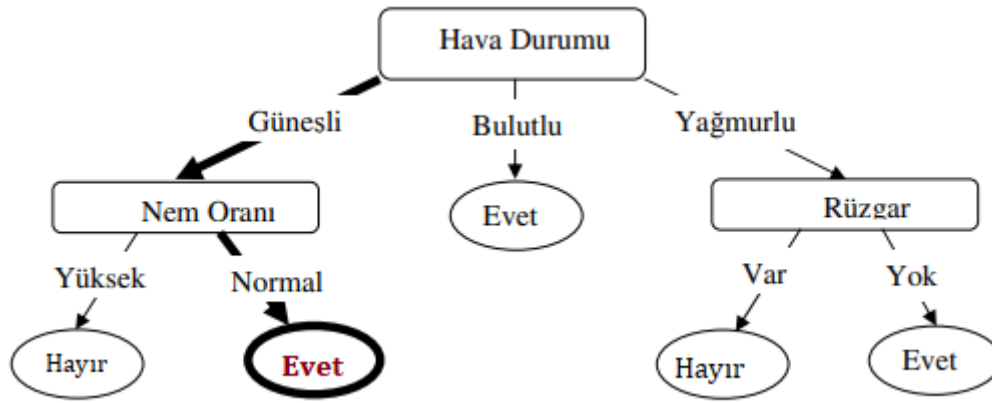
Alt veri kümelerindeki kayıtlar aynı sınıftan olana kadar, her bir veri kümesi için bu işlemler tekrarlanır. Hesaplamalar sonucunda oluşan karar ağacı aşağıda gösterilmiştir.



Oyun oyna niteliği boş olan kayıtların, hangi sınıfa ait olduğu bu karar ağacı kullanılarak tahmin edilebilir.

Hava Durumu	Sıcaklık	Nem Oranı	Rüzgar	BALIK TUTMA
güneşli	sıcak	normal	yok	?

Yukarıdaki nitelik değerlerine sahip bir kaydın sınıflandırılması şu şekilde gerçekleştirilir; ağacın kök düğümünün hangi nitelik olduğu belirlenir. Burada kök düğüm hava durumu niteliğidir. Örnek kayıttaki hava durumu alanı “güneşli” değerine sahiptir. Karar ağacında “güneşli” olarak adlandırılan dal seçilerek o koldan dallanır. Sıradaki düğüm yine bir test düğümüdür. Nem oranı değerine göre kaydın ne tarafa dallanacağını belirler. Örnek kayıttaki nem oranı alanının değeri “normal” olduğundan ağaçtaki “normal” koluna dallanır. Ulaşılan düğüm bir yaprak düğüm olduğundan sınıf tespit edilmiş olur. Yaprak düğümün adı kaydın hangi sınıfa ait olduğunu belirler. Kayıttaki oyun oyna alanı yaprak düğümün adı olan “Evet” olarak doldurulur. Böylece tahmin işlemi gerçekleştirilmiş olur ve aşağıdaki sonuç elde edilmiş olur.



Hava Durumu	Sıcaklık	Nem Oranı	Rüzgar	Balık tutma
güneşli	sıcak	normal	yok	EVET

C4.5 algoritması

C4.5 Algoritması ilk olarak 1993 yılında Quinlan tarafından ortaya atılmıştır. ID3 Algoritmasının detaylandırılmış halidir. ID3 algoritması sadece kategorik verilerle uğraşırken C4.5 algoritması sayısal değişkenlere de uyarlanabilir (Özkan, 2008). Ayrıca C4.5 algoritması karar ağaçları oluştururken kayıp verilerle uğraşmaz. Kazanım oranını hesaplarken verileri eksik olmayan kayıtları kullanarak karar ağacı oluşturulur. Bu algoritma kayıp verileri diğer veri ve değişkenleri kullanarak kazanç oranı hesaplamada kullanır. Bu sayede daha duyarlı ve daha anlamlı kurallar üreten ağaç elde edilebilir. C4.5 algoritmasını ID3 algoritmasından ayıran bir diğer farklılık ise ağacın budanması işlemidir. Budama işlemi için iki ayrı yöntem kullanılmaktadır, bunlar;

Alt ağaç yerleştirilmesi: ağaç içerisindeki alt ağaçlar yapraklara dönüştürülür. Ancak, bu değişimin yapılabilmesi için konulacak yaprak hatasının, alt ağacın hatasından düşük olması gerekir.

Alt ağaç yukarı çekilmesi: bir alt ağacın bu ağacı en çok kullanan ağacın yerini almasıdır. Ancak hata oranının bu işlem sonucunda düşmesi gerekir.

C5.0 algoritması

C5.0 algoritması Clementine, Modeler ve RuleQuest vb. yaygın olarak kullanılan veri madenciliği yazılımlarında kullanılan C4.5 algoritmasının geliştirilmiş ticari bir versiyonudur ve büyük veritabanların da kullanılması hedeflenmiştir. C4.5 algoritmasına göre temel farklılığı; hafıza kullanımında %90 oranında daha az bellek kullanması, hız konusunda ise 5,7 ila 240 kez arasında daha hızlı çalıştığı ve daha kesin kurallar ürettiği gösterilebilir. Bazı veri kümelerinde bulunan hata oranının C4.5 ile bulunan yarısından daha az olduğu belirtilmiştir. Ayrıca C5.0 algoritması daha küçük karar ağaçları oluşturabilir, verinin ayrıştırılması ve kirli verilerin iyileştirilmesinde daha etkilidir

CART algoritması

Literatürde CART ya da C&RT olarak kısaltılan algoritmanın tam açılımı Classification and Regression Trees Türkçe karşılığı ise Sınıflandırma ve Regresyon Ağaçları olarak tanımlanır. C&RT algoritması ilk olarak Leo ve arkadaşları tarafından 1984 yılında yazıldı. C&RT algoritması kategorik veya sayısal değişkenlerle çalışabilme özelliğine sahiptir. Cevap değişken kategorik ise sınıflandırma ağacı (Classification Tree), sürekli ise regresyon ağacı (Regression Tree) olarak adlandırılır, uygulamalarda genel olarak C&RT şeklinde ifade edilmektedir. Bu algoritma ikili karar ağaçları oluşturur. Her aşamada düğümlerin kendinden daha homojen alt dallara ayrılması sağlanmaktadır. Ayrılma kriteri için geliştirilmiş yöntemler bulunmaktadır. Bunlar arasında gini, twoin, sıralı twoin, simetrik gini, en küçük kareler sapması (Least Squared Deviation) gibi yöntemler kullanılmaktadır. Bu yöntemlerin kullanımı sınıf hedeflerin sürekli ya da kategorik olmasına göre belirlenmektedir. En yaygın olarak kullanılan ayırma kriteri Gini ve Twoin kuralıdır. Veri seti üzerinde kök düğümden başlanarak dallandırmalar yapılır ve durdurma kuralı olmadan yapraklara kadar ulaşılır böylece ağaç en yüksek büyüklüğüne ulaşır. C&RT algoritması çok karmaşık bir ağaç oluşturduktan sonra, çapraz geçerlilik (cross validation) ve test (testing) kümesi sınaması işlemi sonuçlarını dikkate alarak en uygun

UİST468.MAKİNE ÖĞRENMESİ

ağaca ulaşır. Daha sonra karar ağacı, test veri kümesi üzerindeki budanmış versiyonlarının başarılarını temel alarak geriye doğru budama işlemini gerçekleştirir. Karar ağacı, yeni ve daha önceden görülmemiş veriyi, çapraz geçerlilik sınanması ile seçer. C&RT algoritmasında, vekil değişkenler kullanılmaktadır yani kayıp olan gözlem değeri için vekil belirlemekte ve kayıp değerleri bu vekil değişken ile doldurmaktadır

Gini Endeksi, CART algoritmasında sınıflandırma görevleri için bir metriktir. Her sınıfın kare olasılıklarının toplamını saklar. Aşağıda gösterildiği gibi formüle edebiliriz.

$$\text{Gini} = 1 - \sum (P_i)^2$$

Hava durumu, nominal bir özelliktir. Güneşli, bulutlu veya yağmurlu olabilir. Hava durumu özelliği için son kararlar aşağıdaki gibi özetlenebilir:

Hava durumu	Evet	Hayır	Toplam
Güneşli	2	3	5
Bulutlu	4	0	4
Yağmurlu	3	2	5

$$\text{Gini (Hava durumu=Güneşli)} = 1 - (2/5)^2 - (3/5)^2 = 1 - 0.16 - 0.36 = 0.48$$

$$\text{Gini (Hava durumu =Bulutlu)} = 1 - (4/4)^2 - (0/4)^2 = 0$$

$$\text{Gini (Hava durumu =Yağmurlu)} = 1 - (3/5)^2 - (2/5)^2 = 1 - 0.36 - 0.16 = 0.48$$

$$\text{Gini (Hava durumu)} = (5/14) \times 0.48 + (4/14) \times 0 + (5/14) \times 0.48 = 0.171 + 0 + 0.171 = 0.342$$

UİST468.MAKİNE ÖĞRENMESİ

Sıcaklık:

Benzer şekilde, sıcaklık nominal bir özelliktir ve 3 farklı değere sahip olabilir: soğuk, sıcak ve hafif. Sıcaklık özelliği ile ilgili kararlar aşağıdaki tabloda özetlenir.

Sıcaklık	Evet	Hayır	Toplam
Sıcak	2	2	4
Soğuk	3	1	4
Ilık	4	2	6

$$\text{Gini (Sıcaklık=Sıcak)} = 1 - (2/4)^2 - (2/4)^2 = 0.5$$

$$\text{Gini (Sıcaklık=Soğuk)} = 1 - (3/4)^2 - (1/4)^2 = 1 - 0.5625 - 0.0625 = 0.375$$

$$\text{Gini (Sıcaklık=Ilık)} = 1 - (4/6)^2 - (2/6)^2 = 1 - 0.444 - 0.111 = 0.445$$

$$\text{Gini (Sıcaklık)} = (4/14) \times 0.5 + (4/14) \times 0.375 + (6/14) \times 0.445 = 0.142 + 0.107 + 0.190 = 0.439$$

Nem:

Nem, ikili sınıfın bir özelliğidir. Yüksek veya normal olabilir.

Nem	Evet	Hayır	Toplam
Yüksek	3	4	7
Normal	6	1	7

$$\text{Gini(Nem =yüksek)} = 1 - (3/7)^2 - (4/7)^2 = 1 - 0.183 - 0.326 = 0.489$$

$$\text{Gini(nem =Normal)} = 1 - (6/7)^2 - (1/7)^2 = 1 - 0.734 - 0.02 = 0.244$$

$$\text{Gini(Nem)} = (7/14) \times 0.489 + (7/14) \times 0.244 = 0.367$$

UİST468.MAKİNE ÖĞRENMESİ

Rüzgar:

Rüzgar, ikili sınıfın bir özelliğidir. Var veya yok olabilir.

Nem	Evet	Hayır	Toplam
Yok	6	2	8
Var	3	3	6

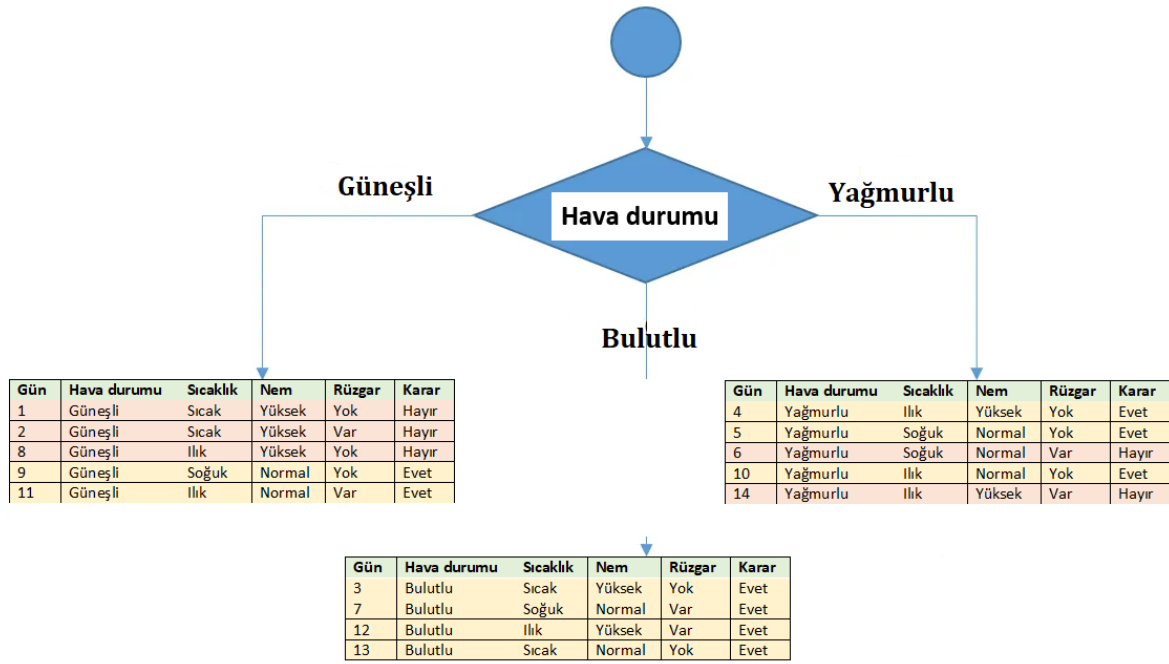
$$\text{Gini(Rüzgar=Yok)} = 1 - (6/8)^2 - (2/8)^2 = 1 - 0.5625 - 0.062 = 0.375$$

$$\text{Gini(Rüzgar=Var)} = 1 - (3/6)^2 - (3/6)^2 = 1 - 0.25 - 0.25 = 0.5$$

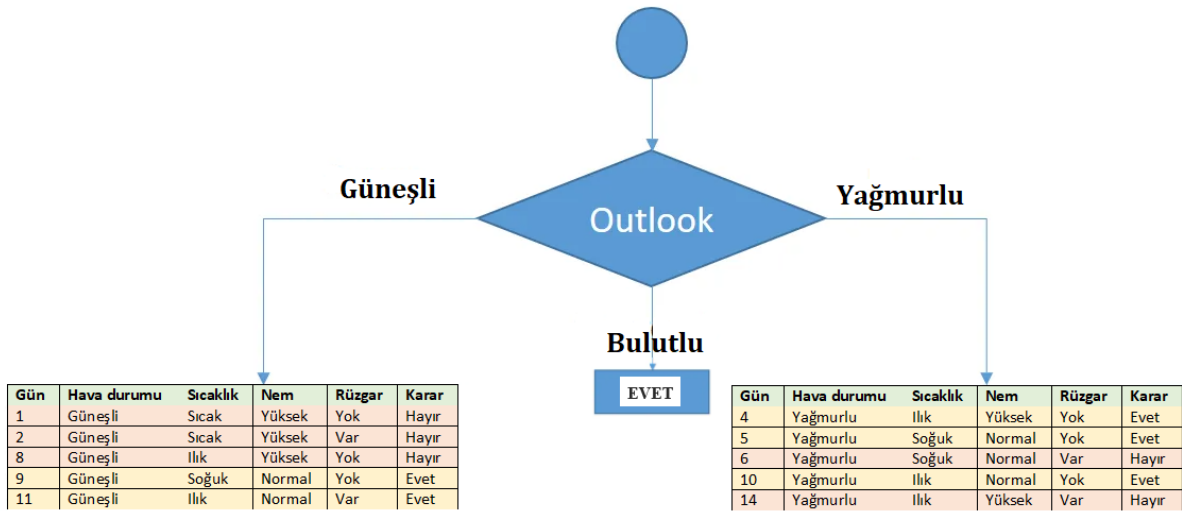
$$\text{Gini(Rüzgar)} = (8/14) \times 0.375 + (6/14) \times 0.5 = 0.428$$

Nitelik	Gini İndeksi
Hava durumu	0,342
Sıcaklık	0,439
Nem	0,367
Rüzgar	0,428

UİST468.MAKİNE ÖĞRENMESİ



Bulutlu yapraktaki alt veri kümesinin sadece Evet kararlarına sahip olduğunu fark edebilirsiniz. Bu, bulutlu yaprağın bittiği anlamına gelir.



Aşağıdaki adımlarda bu alt veri kümelerine aynı ilkeleri uygulayacağız.

Hava durumu GÜNEŞLİ:

UİST468.MAKİNE ÖĞRENMESİ

Güneşli hava durumu için alt veri kümesine odaklanın. Sırasıyla sıcaklık, nem ve rüzgar özellikleri için gini indeks puanlarını bulmamız gerekiyor.

Gün	Hava durumu	Sıcaklık	Nem	Rüzgar	Karar
1	Güneşli	Sıcak	Yüksek	Yok	Hayır
2	Güneşli	Sıcak	Yüksek	Var	Hayır
8	Güneşli	Ilık	Yüksek	Yok	Hayır
9	Güneşli	Soğuk	Normal	Yok	Evet
11	Güneşli	Ilık	Normal	Var	Evet

Güneşli hava durumu için sıcaklık Gini katsayısı:

Sıcaklık	Evet	Hayır	Toplam
Sıcak	0	2	2
Soğuk	1	0	1
Ilık	1	1	2

$$\text{Gini}(\text{Hava durumu=Güneşli ve Sıcaklık=Sıcak}) = 1 - (0/2)^2 - (2/2)^2 = 0$$

$$\text{Gini}(\text{Hava durumu=Güneşli ve Sıcaklık=Soğuk}) = 1 - (1/1)^2 - (0/1)^2 = 0$$

$$\text{Gini}(\text{Hava durumu=Güneşli ve Sıcaklık=Ilık}) = 1 - (1/2)^2 - (1/2)^2 = 1 - 0.25 - 0.25 = 0.5$$

$$\text{Gini}(\text{Hava durumu=Güneşli ve Sıcaklık}) = (2/5) \times 0 + (1/5) \times 0 + (2/5) \times 0.5 = 0.2$$

Güneşli hava durumu için Nem Gini katsayısı:

UİST468.MAKİNE ÖĞRENMESİ

Nem	Evet	Hayır	Toplam
Yüksek	0	3	3
Normal	2	0	2

$$\text{Gini}(\text{Hava durumu=Güneşli ve Nem=Yüksek}) = 1 - (0/3)^2 - (3/3)^2 = 0$$

$$\text{Gini}(\text{Hava durumu=Güneşli ve Nem=Normal}) = 1 - (2/2)^2 - (0/2)^2 = 0$$

$$\text{Gini}(\text{Hava durumu=Güneşli ve Nem}) = (3/5) \times 0 + (2/5) \times 0 = 0$$

Güneşli hava durumu için Rüzgar Gini katsayısı:

Nem	Evet	Hayır	Toplam
Yok	1	2	3
Var	1	1	2

$$\text{Gini}(\text{Hava durumu = Güneşli ve Rüzgar=Yok}) = 1 - (1/3)^2 - (2/3)^2 = 0.266$$

$$\text{Gini}(\text{Hava durumu = Güneşli ve Rüzgar=Var}) = 1 - (1/2)^2 - (1/2)^2 = 0.2$$

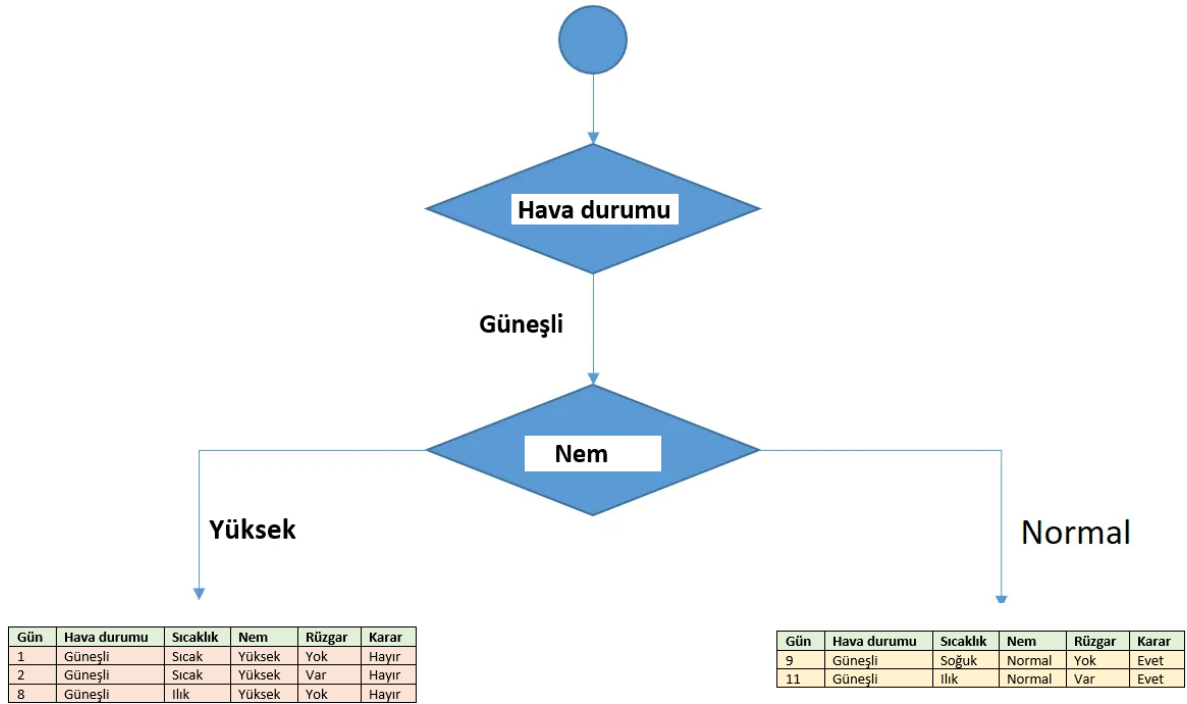
$$\text{Gini}(\text{Hava durumu = Güneşli ve Rüzgar}) = (3/5) \times 0.266 + (2/5) \times 0.2 = 0.466$$

Hava durumu güneşli olduğunda özellik için gini indeksi puanlarını hesapladık. Kazanan nemdir, çünkü en düşük değere sahiptir.

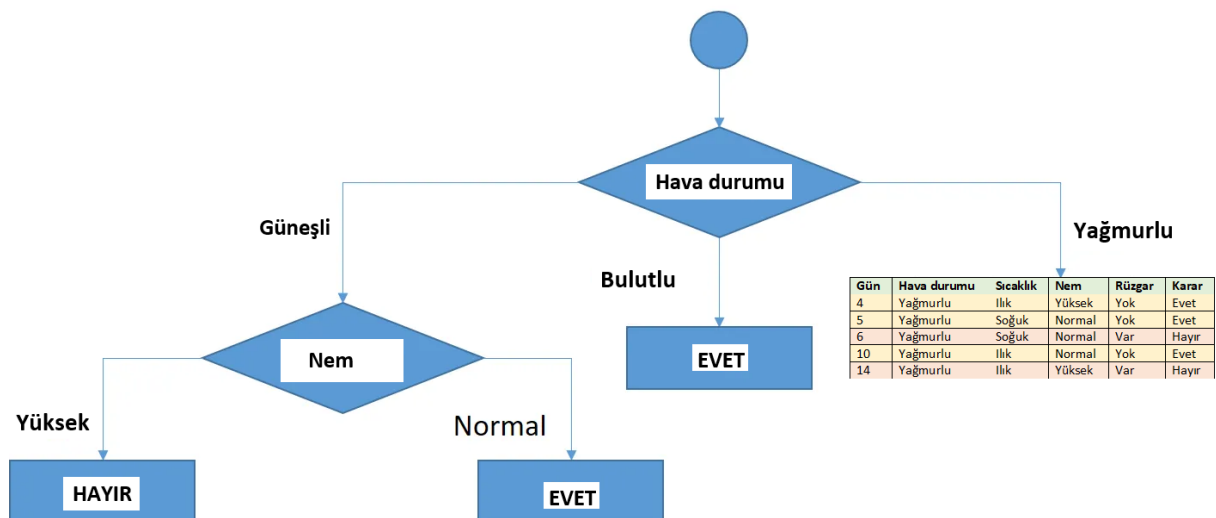
Nitelik	Gini İndeksi
---------	--------------

UİST468.MAKİNE ÖĞRENMESİ

Hava durumu	0,342
Sıcaklık	0,439
Nem	0,367
Rüzgar	0,428



Görüldüğü gibi, yüksek nem ve güneşli bir hava durumu için her zaman bir çözüm yoktur. Öte yandan, karar her zaman normal nem ve güneşli görünüm için Evet olacaktır. Dolayısıyla, bu dal sona ermiştir.



UİST468.MAKİNE ÖĞRENMESİ

Hava durumu YAĞMURLU:

Gün	Hava durumu	Sıcaklık	Nem	Rüzgar	Karar
4	Yağmurlu	Ilık	Yüksek	Yok	Evet
5	Yağmurlu	Soğuk	Normal	Yok	Evet
6	Yağmurlu	Soğuk	Normal	Var	Hayır
10	Yağmurlu	Ilık	Normal	Yok	Evet
14	Yağmurlu	Ilık	Yüksek	Var	Hayır

Hava durumu yağmurlu olduğunda sıcaklık, nem ve rüzgar özellikleri için gini indeksi puanlarını hesaplayacağız.

YAĞMURLU hava durumu için sıcaklık Gini katsayısı:

Sıcaklık	Evet	Hayır	Toplam
Sıcak	0	0	0
Soğuk	1	1	2
Ilık	2	1	3

$$\text{Gini(Hava durumu = yağmurlu ve Sıcaklık= Soğuk)} = 1 - (1/2)^2 - (1/2)^2 = 0.5$$

$$\text{Gini(Hava durumu = yağmurlu ve Sıcaklık= Ilık)} = 1 - (2/3)^2 - (1/3)^2 = 0.444$$

$$\text{Gini(Hava durumu = Yağmurlu ve sıcaklık)} = (2/5) \times 0.5 + (3/5) \times 0.444 = 0.466$$

YAĞMURLU hava durumu için Nem Gini katsayısı:

Nem	Evet	Hayır	Toplam
Yüksek	1	1	2
Normal	2	1	3

UİST468.MAKİNE ÖĞRENMESİ

$$\text{Gini(hava durumu = Yağmurlu ve Nem= Yüksek)} = 1 - (1/2)^2 - (1/2)^2 = 0.5$$

$$\text{Gini(hava durumu = Yağmurlu ve Nem=Normal)} = 1 - (2/3)^2 - (1/3)^2 = 0.444$$

$$\text{Gini(hava durumu = Yağmurlu ve Nem)} = (2/5) \times 0.5 + (3/5) \times 0.444 = 0.466$$

YAĞMURLU hava durumu için rüzgar Gini katsayısı:

Rüzgar	Evet	Hayır	Toplam
Yok	3	0	3
Var	0	2	2

$$\text{Gini(hava durumu = Yağmur ve Rüzgar=yok)} = 1 - (3/3)^2 - (0/3)^2 = 0$$

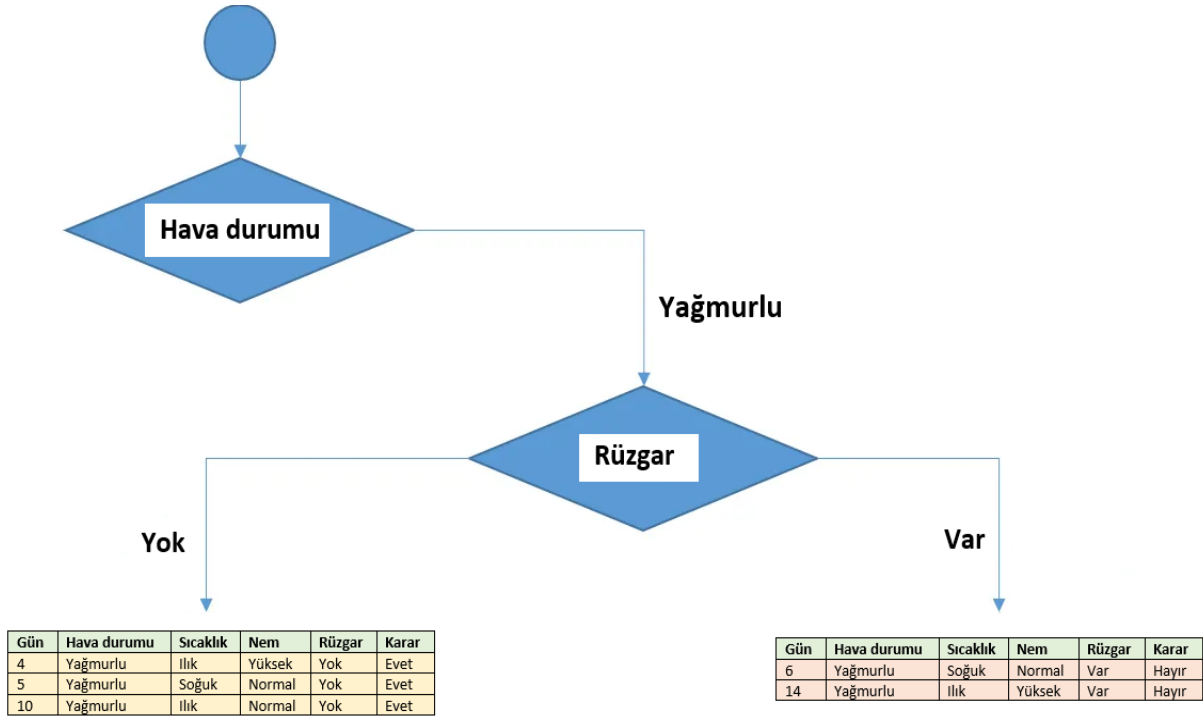
$$\text{Gini(hava durumu = Yağmur ve Rüzgar=var)} = 1 - (0/2)^2 - (2/2)^2 = 0$$

$$\text{Gini(hava durumu = Yağmur ve Rüzgar)} = (3/5) \times 0 + (2/5) \times 0 = 0$$

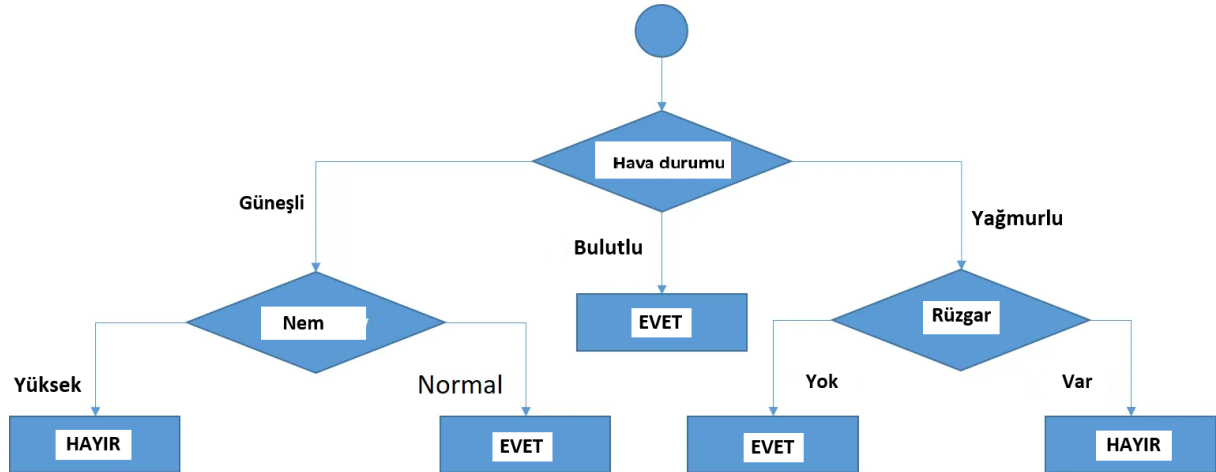
Kazanan, yağmurlu hava durumu için rüzgar özelliğidir, çünkü özelliklerde minimum gini indeks puanına sahiptir.

Nitelik	Gini İndeksi
Sıcaklık	0,466
Nem	0,466
Rüzgar	0

Yağmurlu hava durumu dalı için rüzgar özelliği karar ağacına konup ve yeni alt veri kümelerinin izlenmesi gerekir



Görüldüğü gibi, rüzgar olmadığında karar her zaman evet. Öte yandan, rüzgar var ise karar her zaman hayırdır. Bu, bu dalın bittiği anlamına gelmektedir.



Karar kuralları:

K1: Eğer (hava durumu=Güneşli) ve (Nem=Yüksek) ise balık tutma= HAYIR

K2: Eğer (hava durumu=Güneşli) ve (Nem=Normal) ise balık tutma= EVET

K3: Eğer (hava durumu=Bulutlu) ise balık tutma= EVET

K4: Eğer (hava durumu=Yağmurlu) ve (Rüzgar=yok) ise balık tutma= EVET

K5: Eğer (hava durumu=Yağmurlu) ve (Rüzgar=var) ise balık tutma= HAYIR.

Karar Ağaçlarının Avantajları:

- Anlaması ve yorumlaması kolaydır. Kullanılan ağaç yapılar görselleştirilebilir.
- Az oranda bir veri hazırlığına ihtiyaç duyar. Fakat unutulmamalıdır ki bu model kayıp değerleri desteklememektedir.
- Kullanılan ağacın maliyeti, ağacı eğitmek için kullanılan veri noktalarının sayısıyla logaritmiktir.
- Hem sayısal hem de kategorik verileri işleyebilir.
- Çok çıktılı problemleri ele alabilmektedirler.
- İstatistiksel testler kullanılarak bir modelin doğrulanması mümkündür.
- Karar ağaçları, parametrik olmayan bir yöntem olarak düşünülebilir. Yani uzay dağılımı ve sınıflandırma yapısı hakkında bir yaklaşıma sahip değildir.

Karar Ağaçlarının Dezavantajları:

- Veriyi iyi bir şekilde açıklamayan aşırı karmaşık ağaçlar üretilebilir. Bu durumda ağaç dallanması takip edilemeyebilir.

UİST468.MAKİNE ÖĞRENMESİ

- Ezbere öğrenme yaşanabilir (“over-fitting”). Bu problemin çözümü için model parametrelere kısıtlamalar ve budama gibi yöntemler kullanılabilir. Budama işlemi, az sayıda nesneyi barındıran yaprak düğümlerin karar ağacı grafiğinden atılmasını ifade etmektedir

Stratejik bölünmeler yapma kararı, bir ağacın doğruluğunu büyük ölçüde etkiler. Karar kriterleri sınıflandırma ve regresyon ağaçları için farklıdır. Karar ağaçları, bir düğümü iki veya daha fazla alt düğüme bölmeye karar vermek için birden çok algoritma kullanır. Alt düğümlerin oluşturulması, ortaya çıkan alt düğümlerin homojenliğini artırır. Başka bir deyişle, düğümün saflığının hedef değışkene göre arttığını söyleyebiliriz. Karar ağacı, mevcut tüm değışkenlerdeki düğümleri böler ve daha sonra en homojen alt düğümlerle sonuçlanan bölünmeyi seçer.

1.KARAR AĞAÇLARI ALGORİTMASI UYGULAMASI

Klasik makine öğrenmesi sınıflama problemlerinde kullanılan hava durumuna göre hafta sonu balık tutmaya gidip gitmeme kararı veri seti kullanılacaktır. Veri seti aşağıdaki gibidir.

HAVA_DURUMU	SICAKLIK	NEM	RÜZGAR	KARAR
güneşli	sıcak	yüksek	zayıf	Hayır
güneşli	sıcak	yüksek	kuvvetli	Hayır
bulutlu	sıcak	yüksek	zayıf	Evet
yağmurlu	ılık	yüksek	zayıf	Evet
yağmurlu	soğuk	normal	zayıf	Evet
yağmurlu	soğuk	normal	kuvvetli	Hayır
bulutlu	soğuk	normal	kuvvetli	Evet
güneşli	ılık	yüksek	kuvvetli	Hayır
güneşli	soğuk	normal	zayıf	Evet
yağmurlu	ılık	normal	zayıf	Evet
güneşli	ılık	normal	kuvvetli	Evet
bulutlu	ılık	yüksek	kuvvetli	Evet
bulutlu	sıcak	normal	zayıf	Evet
yağmurlu	ılık	yüksek	kuvvetli	Hayır

PROBLEM:

Hava durumu veri setini ve karar ağaçları (ID3) algoritmasını kullanarak hafta sonu balığa gidip gitmeme tahmini yapan karar kuralları oluşturmak.

Gerekli kütüphaneler ve veri setinin girilmesi için aşağıdaki kodlar kullanılır.

```
import pandas as pd
import numpy as np
df = pd.DataFrame()
df['HAVA_DURUMU'] = [
    'güneşli', 'güneşli', 'bulutlu', 'yağmurlu', 'yağmurlu', 'yağmurlu',
    'bulutlu', 'güneşli', 'güneşli', 'yağmurlu', 'güneşli',
    'bulutlu', 'bulutlu', 'yağmurlu'
]
df['SICAKLIK'] = [
    'sıcak', 'sıcak', 'sıcak', 'ılık', 'soğuk', 'soğuk', 'soğuk',
    'ılık', 'soğuk', 'ılık', 'ılık', 'ılık', 'sıcak', 'ılık',
]
df['NEM'] = [
    'yüksek', 'yüksek', 'yüksek', 'yüksek', 'normal', 'normal', 'normal',
    'yüksek', 'normal', 'normal', 'normal', 'yüksek', 'normal', 'yüksek'
]
df['RÜZGAR'] = [
    'zayıf', 'kuvvetli', 'zayıf', 'zayıf', 'zayıf', 'kuvvetli', 'kuvvetli', 'kuvvetli',
    'zayıf', 'zayıf', 'kuvvetli', 'kuvvetli', 'zayıf', 'kuvvetli'
]
df['KARAR'] = [
    'Hayır', 'Hayır', 'Evet', 'Evet', 'Evet', 'Hayır', 'Evet', 'Hayır', 'Evet', 'Evet',
    'Evet', 'Evet', 'Evet', 'Hayır'
]
```

df

	HAVA_DURUMU	SICAKLIK	NEM	RÜZGAR	KARAR
0	güneşli	sıcak	yüksek	zayıf	Hayır
1	güneşli	sıcak	yüksek	kuvvetli	Hayır
2	bulutlu	sıcak	yüksek	zayıf	Evet
3	yağmurlu	ılık	yüksek	zayıf	Evet
4	yağmurlu	soğuk	normal	zayıf	Evet
5	yağmurlu	soğuk	normal	kuvvetli	Hayır
6	bulutlu	soğuk	normal	kuvvetli	Evet
7	güneşli	ılık	yüksek	kuvvetli	Hayır
8	güneşli	soğuk	normal	zayıf	Evet
9	yağmurlu	ılık	normal	zayıf	Evet
10	güneşli	ılık	normal	kuvvetli	Evet
11	bulutlu	ılık	yüksek	kuvvetli	Evet
12	bulutlu	sıcak	normal	zayıf	Evet
13	yağmurlu	ılık	yüksek	kuvvetli	Hayır

Veri seti küçük olduğundan eğitim ve test verisi olarak ayırım yapılmadan tüm veri eğitim verisi olarak kullanılacaktır.

1. adım: İlk adım aşağıdaki kodlar yardımıyla KARAR hedef değişkeni için entropi değerini hesaplamaktır.

```
# EVET OLASILIĞINI HESAPLAMA
p_evet = len(df.loc[df.KARAR == 'Evet']) / len(df)
# HAYIR OLASILIĞINI HESAPLAMA
p_hayır = len(df.loc[df.KARAR == 'Hayır']) / len(df)
entropi_karar = -p_hayır * np.log2(p_hayır) - p_evet * np.log2(p_evet)
print(f'H(S) = {entropi_karar:0.4f}')
```

$$H(S) = 0.9403$$

Bu hesaplama, bir entropi karar fonksiyonuna dönüştürmek için aşağıdaki kodlarla tekrar edilir.

```
def f_entropi_karar(data):
    p_evet = len(data.loc[data.KARAR == 'Evet']) / len(data)
    p_hayır = len(data.loc[data.KARAR == 'Hayır']) / len(data)
    return -p_hayır * np.log2(p_hayır) - p_evet * np.log2(p_evet)
```

UİST468.MAKİNE ÖĞRENMESİ

2.adım. Bir sonraki adım, hangi öz niteliğin Hava durumu, sıcaklık, nem ve rüzgârdan en yüksek bilgi kazancını sağladığını hesaplamaktır. Hava durumu parametresinden başlayarak, güneşli, bulutlu ve yağmurlu koşullar verilen her kararın olasılığı aşağıdaki fonksiyon ve kodlar kullanarak hesaplanır.

$$BK(S,a) = H(S) - H(S|a) = H(S) - \sum_{t \in T} p(t)H(t)$$

Pandas groupby metodu kullanılarak yukarıdaki eşitlik elde edilebilir.

```
Bilgi_kazancı_karar_HAVA_DURUMU = entropi_karar # H(S)
# Toplamı yazdırmak için bir string oluşturma
toplam = 'Kazanç(KARAR, HAVA_DURUMU) = Entropi(KARAR)'
# Hava durumu değerleri için iteratif işlemlerle entropi ve olasıkları hesaplama
for isim, HAVA_DURUMU in df.groupby('HAVA_DURUMU'):
    sayı_evet = len(HAVA_DURUMU.loc[HAVA_DURUMU.KARAR == 'Evet'])
    sayı_hayır = len(HAVA_DURUMU.loc[HAVA_DURUMU.KARAR != 'Hayır'])
    sayı_HAVA_DURUMU = len(HAVA_DURUMU)
    print(f'p(KARAR=Evet|HAVA_DURUMU={isim}) = {sayı_evet}/{sayı_HAVA_DURUMU}')
    print(f'p(KARAR=Hayır|HAVA_DURUMU={isim}) = {sayı_hayır}/{sayı_HAVA_DURUMU}')
    print(f'p(KARAR|HAVA_DURUMU={isim}) = {sayı_HAVA_DURUMU}/{len(df)}')
    print(f'Entropi(KARAR|HAVA_DURUMU={isim}) = '\
          f'-{sayı_evet}/{sayı_HAVA_DURUMU}.log2({sayı_evet}/{sayı_HAVA_DURUMU}) - '\
          f'{sayı_hayır}/{sayı_HAVA_DURUMU}.log2({sayı_hayır}/{sayı_HAVA_DURUMU})')
    entropi_KARAR_HAVA_DURUMU = 0

# SIFIRIN LOGARİTMASI ALINAMADIĞI DURUMLAR İÇİN
if sayı_evet != 0:
    entropi_KARAR_HAVA_DURUMU -= (sayı_evet / sayı_HAVA_DURUMU) \
        * np.log2(sayı_evet / sayı_HAVA_DURUMU)
if sayı_hayır != 0:
    entropi_KARAR_HAVA_DURUMU -= (sayı_hayır / sayı_HAVA_DURUMU) \
```

UIST468.MAKİNE ÖĞRENMESİ

```
* np.log2(sayı_hayır/ sayı_HAVA_DURUMU)

Bilgi_kazancı_karar_HAVA_DURUMU -= (sayı_HAVA_DURUMU / len(df)) *
entropi_KARAR_HAVA_DURUMU
print()
toplam += f' - p(KARAR|HAVA_DURUMU={isim}).'
toplam += f'Entropi(KARAR|HAVA_DURUMU={isim})'
print(toplam)
print(f'Kazanç(KARAR, HAVA_DURUMU) = {Bilgi_kazancı_karar_HAVA_DURUMU:0.4f}')
```

```
p(KARAR=Evet|HAVA_DURUMU=bulutlu) = 4/4
p(KARAR=Hayır|HAVA_DURUMU=bulutlu) = 4/4
p(KARAR|HAVA_DURUMU=bulutlu) = 4/14
Entropi(KARAR|HAVA_DURUMU=bulutlu) = -4/4.log2(4/4) - 4/4.log2(4/4)

p(KARAR=Evet|HAVA_DURUMU=güneşli) = 2/5
p(KARAR=Hayır|HAVA_DURUMU=güneşli) = 2/5
p(KARAR|HAVA_DURUMU=güneşli) = 5/14
Entropi(KARAR|HAVA_DURUMU=güneşli) = -2/5.log2(2/5) - 2/5.log2(2/5)

p(KARAR=Evet|HAVA_DURUMU=yağmurlu) = 3/5
p(KARAR=Hayır|HAVA_DURUMU=yağmurlu) = 3/5
p(KARAR|HAVA_DURUMU=yağmurlu) = 5/14
Entropi(KARAR|HAVA_DURUMU=yağmurlu) = -3/5.log2(3/5) - 3/5.log2(3/5)

Kazanç(KARAR, HAVA_DURUMU) = Entropi(KARAR) - p(KARAR|HAVA_DURUMU=bulutlu).Entropi(KARAR|HAVA_DURUMU=bulutlu) - p(KARAR|HAVA_DURUMU=güneşli).Entropi(KARAR|HAVA_DURUMU=güneşli) - p(KARAR|HAVA_DURUMU=yağmurlu).Entropi(KARAR|HAVA_DURUMU=yağmurlu)
Kazanç(KARAR, HAVA_DURUMU) = 0.2467
```

3.ADIM. Yukarıdaki işlemi birkaç kez tekrarlayarak bir fonksiyon haline getirilmesi gerekir.

```
def BK(data, column, ent_KARAR=entropi_karar):
    BK_KARAR = ent_KARAR
    for isim, temp in data.groupby(column):
        p_evet = len(temp.loc[temp.KARAR == 'Evet']) / len(temp)
        p_hayır = len(temp.loc[temp.KARAR != 'Evet']) / len(temp)
        entropi_karar = 0
        if p_evet != 0:
            entropi_karar -= (p_evet) * np.log2(p_evet)
        if p_hayır != 0:
            entropi_karar -= (p_hayır) * np.log2(p_hayır)
        BK_KARAR -= (len(temp) / len(df)) * entropi_karar
    return BK_KARAR
```

4. Her bir öz nitelik için bilgi kazancını hesaplamak için aşağıdaki kod kullanılır.

```
for col in df.columns[:-1]:
    print(f'Kazanç(KARAR, {col}) = {BK(df, col):0.4f}')
```

UİST468.MAKİNE ÖĞRENMESİ

Kazanç(KARAR, HAVA_DURUMU) = 0.2467
 Kazanç(KARAR, SICAKLIK) = 0.0292
 Kazanç(KARAR, NEM) = 0.1518
 Kazanç(KARAR, RÜZGAR) = 0.1518

5. Yukarıdaki sonuç karar ağacı için ilk kararın verilmesini sağlar. Kök olarak enyüksek bilgi kazancını veren hava durumu üzerine veriyi böleceğiz.

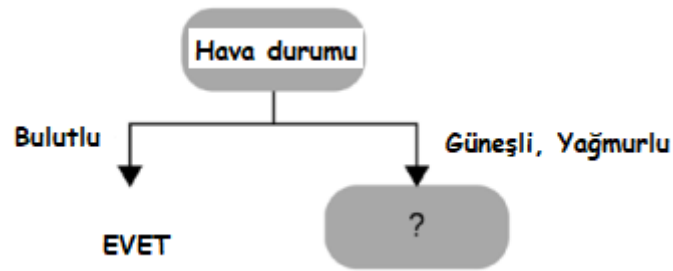
```
for isim, temp in df.groupby('HAVA_DURUMU'):
    print('-' * 15)
    print(isim)
    print('-' * 15)
    print(temp)
    print('-' * 15)
```

```
-----
bulutlu
-----
   HAVA_DURUMU SICAKLIK    NEM    RÜZGAR  KARAR
2    bulutlu    sıcak  yüksek   zayıf   Evet
6    bulutlu    soğuk   normal  kuvvetli  Evet
11   bulutlu    ılık   yüksek  kuvvetli  Evet
12   bulutlu    sıcak   normal   zayıf   Evet
-----
güneşli
-----
   HAVA_DURUMU SICAKLIK    NEM    RÜZGAR  KARAR
0    güneşli    sıcak  yüksek   zayıf  Hayır
1    güneşli    sıcak  yüksek  kuvvetli  Hayır
7    güneşli    ılık   yüksek  kuvvetli  Hayır
8    güneşli    soğuk   normal   zayıf   Evet
10   güneşli    ılık   normal  kuvvetli  Evet
-----
yağmurlu
-----
   HAVA_DURUMU SICAKLIK    NEM    RÜZGAR  KARAR
3    yağmurlu    ılık   yüksek   zayıf   Evet
4    yağmurlu    soğuk   normal   zayıf   Evet
5    yağmurlu    soğuk   normal  kuvvetli  Hayır
9    yağmurlu    ılık   normal   zayıf   Evet
13   yağmurlu    ılık   yüksek  kuvvetli  Hayır
-----
```

Tüm bulutlu kayıtların bir EVET kararına sahip olduğuna dikkat edin. Bu, karar ağacının ilk sonlandırma yaprağımızı sağlar. Eğer bulutlu ise, o zaman balığa

UİST468.MAKİNE ÖĞRENMESİ

gideceğiz ve eğer yağmurlu veya güneşli ise, o zaman gitmeme şansıda söz konusudur. Karar ağacı şu ana kadar aşağıdaki şekilde temsil edilebilir:



6.adım. Şimdi bu işlemi tekrarlıyoruz, tüm veriler tahsis edilene ve ağacın tüm dalları sona erene kadar bilgi kazancına bölünüyoruz. İlk olarak, artık herhangi bir ek bilgi sağlamadıkları için bulutlu örnekleri çıkarın:

```
df_yeni = df.loc[df.HAVA_DURUMU != 'bulutlu']
df_yeni
```

	HAVA_DURUMU	SICAKLIK	NEM	RÜZGAR	KARAR
0	güneşli	sıcak	yüksek	zayıf	Hayır
1	güneşli	sıcak	yüksek	kuvvetli	Hayır
3	yağmurlu	ılık	yüksek	zayıf	Evet
4	yağmurlu	soğuk	normal	zayıf	Evet
5	yağmurlu	soğuk	normal	kuvvetli	Hayır
7	güneşli	ılık	yüksek	kuvvetli	Hayır
8	güneşli	soğuk	normal	zayıf	Evet
9	yağmurlu	ılık	normal	zayıf	Evet
10	güneşli	ılık	normal	kuvvetli	Evet
13	yağmurlu	ılık	yüksek	kuvvetli	Hayır

7.adım. Şimdi, dikkatimizi güneşli örneklerle çevireceğiz ve güneşli bilgileri bölmenin en iyi yolunu belirlemek için kazanç hesaplamalarını yeniden başlatacağız:

```
df_güneşli = df_next.loc[df_next.HAVA_DURUMU == 'güneşli']
df_güneşli
```

	HAVA_DURUMU	SICAKLIK	NEM	RÜZGAR	KARAR
0	güneşli	sıcak	yüksek	zayıf	Hayır
1	güneşli	sıcak	yüksek	kuvvetli	Hayır
7	güneşli	ılık	yüksek	kuvvetli	Hayır
8	güneşli	soğuk	normal	zayıf	Evet
10	güneşli	ılık	normal	kuvvetli	Evet

8.adım. Güneşli örnekler için aşağıdaki kodla entropiyi tekrar hesaplırsak;

```
entropi_karar = f_entropi_karar(df_güneşli)
entropi_karar
```

0.97095059 değeri elde edilir.

9.adım. Güneşli örnekler için kazanç tekrar hesaplanması için aşağıda kod çalıştırılır.

```
for col in df_güneşli.columns[1:-1]:
    print(f' Kazanç(KARAR, {col}) = {BK(df_güneşli, col, entropi_karar):0.4f}')
```

```
Kazanç(KARAR, SICAKLIK) = 0.8281
Kazanç(KARAR, NEM) = 0.9710
Kazanç(KARAR, RÜZGAR) = 0.6313
```

10.adım. En yüksek kazanç nem olduğu için şimdi veriyi nem özniteliğine göre gruplandırılır.

```
for isim, temp in df_güneşli.groupby('NEM'):
    print('-' * 15)
    print(isim)
    print('-' * 15)
    print(temp)
    print('-' * 15)
```

UİST468.MAKİNE ÖĞRENMESİ

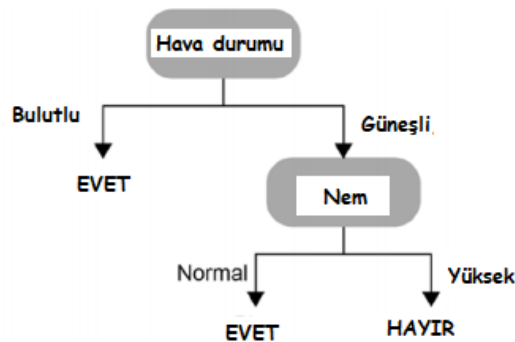
normal					

	HAVA_DURUMU	SICAKLIK	NEM	RÜZGAR	KARAR
8	güneşli	soğuk	normal	zayıf	Evet
10	güneşli	ılık	normal	kuvvetli	Evet

yüksek					

	HAVA_DURUMU	SICAKLIK	NEM	RÜZGAR	KARAR
0	güneşli	sıcak	yüksek	zayıf	Hayır
1	güneşli	sıcak	yüksek	kuvvetli	Hayır
7	güneşli	ılık	yüksek	kuvvetli	Hayır

Burada, nem hem yüksek olduğunda (karar hayır) hem de normal olduğunda (karar evet) tek karar seçeneği olduğundan iki sonlandırma yaprağı söz konusudur. Dolayısıyla güncellenmiş karar ağacı aşağıdaki gibidir.



11.adım. O halde; sınıflandırma gerektiren son veri kümesi yağmurlu hava durumu verileridir. Sadece yağmurlu verilerini ayıklayın ve entropi hesaplamasını yeniden çalıştırın:

```
df_yağmurlu = df_yeni.loc[df_next.HAVA_DURUMU == 'yağmurlu']
entropi_karar = f_entropi_karar(df_yağmurlu)
entropi_karar
```

Yukarıdaki kodlar çalıştırılarak 0.9709505 entropi değeri elde edilir.

UİST468.MAKİNE ÖĞRENMESİ

12. adım. Yağmurlu alt kümesi için kazançlar tekrar hesaplanarak,

```
for col in df_yağmurlu.columns[1:-1]:
    print(f'Kazanç(KARAR, {col}) = {BK(df_yağmurlu, col, entropi_karar):0.4f}')
```

Kazanç(KARAR, SICAKLIK) = 0.6313

Kazanç(KARAR, NEM) = 0.6313

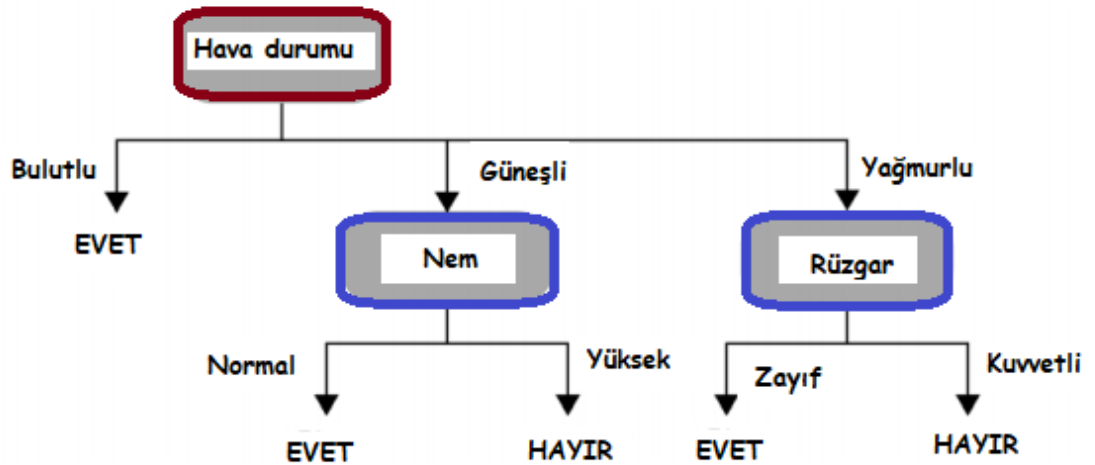
Kazanç(KARAR, RÜZGAR) = 0.9710

13. En yüksek bilgi kazancı Rüzgar özniteliği için çıktığından rüzgar özniteliğine göre kalan bilgi gruplandırılır.

```
for isim, temp in df_yağmurlu.groupby('RÜZGAR'):
    print('-' * 15)
    print(isim)
    print('-' * 15)
    print(temp)
    print('-' * 15)
```

```
-----
kuvvetli
-----
      HAVA_DURUMU SICAKLIK      NEM  RÜZGAR  KARAR
5      yağmurlu   soğuk  normal  kuvvetli  Hayır
13     yağmurlu    ılık  yüksek  kuvvetli  Hayır
-----
zayıf
-----
      HAVA_DURUMU SICAKLIK      NEM  RÜZGAR  KARAR
3      yağmurlu    ılık  yüksek  zayıf    Evet
4      yağmurlu   soğuk  normal  zayıf    Evet
9      yağmurlu    ılık  normal  zayıf    Evet
-----
```

Sonuç olarak, ağacı tamamlamak için gereken tüm sonlandırma yapraklarına sahibiz, çünkü rüzgar bölmesi , hepsi evet veya hayır değerlerini gösteren iki set vermektedir.. Tam karar ağacımız aşağıdaki gibidir:



Böylece, hava durumuna bakarak önümüzdeki hafta balığa çıkıp çıkmama kararını tahmin edebiliriz. Yapılacak işlem, sadece ağacını takip etmektir. Her düğümde yapılan karara bakın ve görünmeyen örnekten gelen verileri uygulayın. Tahmin daha sonra sonlandırma sayfasında belirtilen etiket olarak sona erecektir. Diyelim ki önümüzdeki hafta için hava durumu; yağmurlu, sıcaklık ılık, nem normal ve rüzgar kuvvetli olacaktır. Aşağıdaki karar ağacını takip ederek kararımız hayır olacaktır.

