

KARAR AĞAÇLARI (DECISION TREES)

1. HAZIRLAYANLAR:

2. AHMET TARIK EFE - G181210009
3. NECDET CAN ÖZBEK - G181210111
4. UMMAN CEBE- G181210021
5. ERSİN KARAKULAK - B201210371



İÇİNDEKİLER

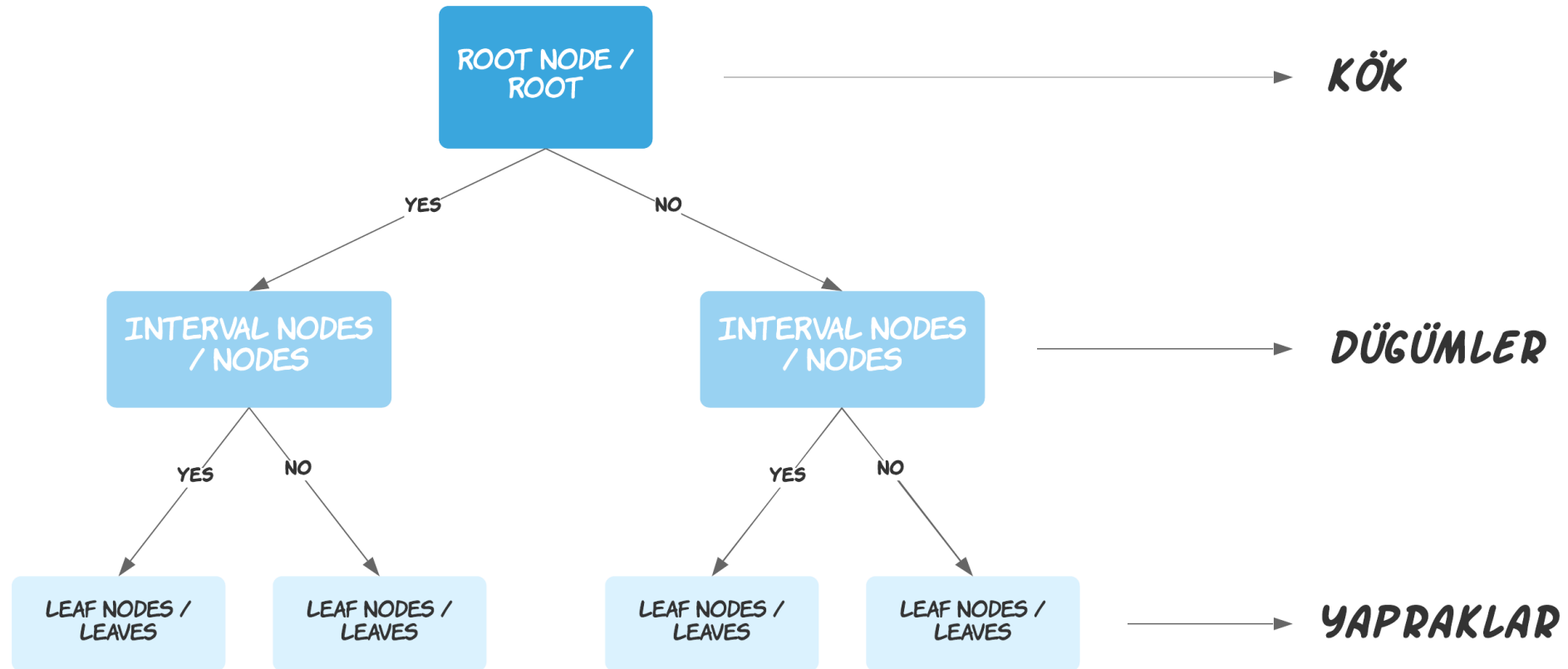
- KARAR AĞACI NEDİR ?
- KARAR AĞAÇLARINA DAİR TEMEL KAVRAMLAR
- KARAR AĞACI ALGORİTMALARI
- KARAR AĞAÇLARININ AVANTAJLARI
- KARAR AĞAÇLARININ YAPAY ZEKA İÇİN ÖNEMİ
- CART VE ID3 ALGORİTMALARI İLE KARAR AĞACI OLUŞTURMA
- KARAR AĞACI ÖRNEKLERİ

KARAR AĞACI NEDİR ?

- Karar ağacının en altında yapraklar (leaf nodes) bulunur. Bu yapraklar bizlere aradığımız sonucu verir. Karar ağacı oluştururken en baştaki kök düğümü belirlemek için önce problem belirlenir daha sonra bu problemdeki en önemli etken belirlenerek kök düğüm bu etken üzerinden oluşturulur. Karar ağacı üzerinde bir problem üzerindeki tüm etmenler dikkate alındığı için o problem hakkında en doğru kararı vermek veya en doğru çıkarımı yapmak sistematik bir şekilde ilerlenebildiği için çok kolay ve uygulanabilirdir.

• KARAR AĞACINI OLUŞTURAN TEMEL ELEMANLAR

- **1-) KÖK DÜĞÜM** : Nihai hedefi veya karar vermeye çalıştığımız problemi temsil eder.
- **2-) DALLAR** : Kökler üzerinden ilerleyen dallar belirli bir karar verirken mevcut olan farklı seçenekleri temsil eder. Genellikle bir ok çizgisi olarak gösterilirler bunun yanı sıra maliyeti ve gerçekleşme olasılığını da içerirler
- **3-) İÇ KARAR DÜĞÜMLERİ** : Giriş verilini test eden soruların sorulduğu ve hangi yöne ilerleneceğini belirleyen düğümlerdir. Kök düğüm üzerinden dallar aracılığıyla oluşurlar.
- **4-) YAPRAK DÜĞÜM** : Ağacın en sonundaki düğümlere yaprak düğüm denir. Yaprak düğümler bir kararı veya sonucu temsil ederler. Karar ağacı üzerinden ilerlenen yol sonucunda yaprak düğümdeki ifade bizim kararımız veya sonucumuzdur.



KARAR AĞAÇLARINA DAİR TEMEL KAVRAMLAR

- **Entropi** : Entropi, işlenmekte olan bilgidaki rastgeleliğin bir ölçüsüdür. Entropi ne kadar yüksekse, değişkene ait bilgiden herhangi bir sonuç çıkarmak o kadar zor olur.
- **Bilgi Kazanımı (Information Gain)**: Bir değişkene ait çıktı değişken üzerindeki açıklayıcılık etkisi düğüm bölünmesine karar vermek için kullanılır.
- **Gini Index** : Gini endeksi, veri kümesindeki bölünmeleri değerlendirmek için kullanılan bir maliyet fonksiyonudur. Her bir sınıfın olasılıklarının karelerinin toplamının birden çıkarılmasıyla hesaplanır. Gini değeri düşük olan nitelik tercih edilir.

GİNİ INDEX , ENTROPİ ve INFORMATION GAIN DEĞERLERİNİN HESAPLANMASI

- 1 – GİNİ INDEX

$$Gini = 1 - \sum_j p_j^2$$

- p_j , j sınıfının gerçekleşme olasılığıdır. Her sınıf için hesaplanır ve çıkan sonuçların karelerinin toplamı birden çıkartılır. Gini değeri 0 ile 1 arasında bir sonuç alır ve sonuç 0'a ne kadar yakınsa o kadar iyi ayırım yapmış olur.

- 2 – ENTROPİ

$$Entropy(D_1) = -\sum_{i=1}^m p_i \log_2 p_i$$

m : Entropisi hesaplanacak durum sayısı

- **P_i** : *i* durumunun olasılığı

3 – INFORMATION GAIN (BİLGİ KAZANCI)

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

Gain(S,A); A niteliğinin S sistemine göre kazancı anlamına gelir. Bu değer sistemin entropisinden, özniteliğin entropi değerinin çıkarılmasıyla elde edilir.

KARAR AĞACI ALGORİTMALARI

- Karar ağacı oluşturmak için kullanılan birçok algoritma vardır. Karar ağacı oluşturmak istediğimizde bir algoritma seçmemiz gerekmektedir. Doğru algoritmanın seçimi model başarısını önemli ölçüde etkilemektedir. Bazı algoritmalar 0 ve 1 değerlerinden oluşan veri setleri üzerinde çalışmaktadır. Bir yapay sinir ağı çalışmasında nitel değişkenin var-yok şeklinde değerlendirilmesi ya da bir karar ağaçları çalışmasında nicel bir değişkenin kategorilere ayrılması modelin performansını arttırmaktadır. Bu örnek durumlardan hareketle, algoritmaların yöntemlere göre avantajları ve dezavantajları mevcuttur.
- Karar ağaçları algoritmaları; AID, CHAID, CART, ID3, C4.5, C5.0, MARS, E-CHAID, SLIQ, SPRINT ve QUEST şeklindedir. Karar ağaçları algoritmaları kök, düğüm ve dallanma kriterlerinin seçiminde izlenen yollar ile birbirinden ayrılır.

Örnekler

Karar Ağacı Algoritması

Karar Ağacı

içerik

Yeni Örnek

sınıflandırma

AID

- 1970'li yılların başlarında Morgan ve Sonquist adlı uzmanlar tarafından ileri sürülen AID algoritması, karar ağacı tabanlı hem ilk algoritma hem de ilk yazılımdır. AID, en iyi tahmini gerçekleştirmeye ve en kuvvetli ilişkiye sahip bağımsız değişkeni bulmaya dayanıyor. AID değişken değerlerine göre ikiye ayrılır. Bu süreç bölünmeler sonlanıncaya kadar devam etmektedir. Bağımsız değişkenlerin türleri sınıflayıcı ve sıralayıcı özellikte olabilmektedir. Sonuçları ikili ağaç yapısında sunan AID'nin regresyon analizinden farkı değişkenlerin arasından anlamlılığı hakkında güvenilir bir bilgi vermemesidir.

CHAID

- Sınıflandırma ve regresyon amacı taşıyan ve istatistik tabanlı olan CHAID algoritması G. V. Kass tarafından 1980’de geliştirilmiştir. CHAID ile oluşturulacak bir regresyon denklemi normal dağılım, doğrusallık ve homojenlik gibi bilinen klasik varsayımlardan ayrı tutulmaktadır. Çünkü güçlü bir iterasyon algoritmasıyla bütün olan ana kütle kararlı alt düğümlere bölünebilmektedir.

CART

- 1984 yılında Breiman ve arkadaşları tarafından çalışmalarında bir karar ağacı algoritması olan CART algoritması kullanılarak, literatüre kazandırılmıştır. Hem sınıflandırma hem de regresyon amacıyla kullanılan CART algoritması entropiye dayalı olup, dallanma kriterinin hesaplanmasında Twoing ve Gini tekniklerinden faydalanmaktadır. CART algoritması her adımda ilgili grubun, kendinden daha homojen olan iki alt gruba ayrılmasını sağlamaktadır. Yani her dal ikili alt gruplara ayrılarak büyümektedir. Ayırma işleminde bağımlı değişken kategorik ise gini veya twoing, sürekli ise en küçük kareler sapmasından yararlanmaktadır.

ID3

- J. Ross Quinlan tarafından 1986 yılında ID3 adlı bir karar ağacı algoritması geliştirilmiştir. Bu algoritma sadece ayırık yerlerde üzerinde çalışabilmektedir. ID3, sayısal öznitelikler veya eksik durumlar işlem yapmamakta ayrıca bir budama işlemi uygulamamaktadır. Temel olarak algoritma kategorik nitelikleri sınıflandırmaktadır.

C4.5

- J. Ross Quinlan yayınladığı kitap ile “ID3 algoritmasının ileri bir sürümü olan C4.5 karar ağacı algoritmasına literatür kazandırmıştır. ID3 algoritmasının eksik yönlerini gidermek üzere Quinlan tarafından, 1993 yılında mevcuttur. ID3’e göre en büyük farkları sayısal ortamda üzerinde çalışabilmesi ve eksik verilerin işleyebilmesinden. C4.5 algoritmasının ID3’ten daha farklı bir ‘kazanım’ oranı içindir. ID3 algoritması kazanç ölçütünü kullanmaktadır. C4.5 algoritması bölünmüş bilgisi kavramıyla kazanç ölçütü seçmek hesaplanan kazanç oranını kullanmaktadır.

C5.0

- J. Ross Quinlan tarafından C4.5 algoritması geliştirilerek, C5 algoritması elde edilmiştir. Kategorik hedef değişkenleri tahmin etmede kullanılan C5.0 algoritmasının C4.5 algoritmasına göre üstünlükleri daha hızlı olması, daha az bellek kullanması, daha kesin kurallar oluşturması, ağacın budanabilmesi, değişkenlerin ve yanlış sınıflandırma türlerinin ağırlıklandırılabilmesi ve ağacın oluşumuna katkı sağlamayan değişkenleri dışlayabilmesidir. Ayrıca, doğruluk oranını artırıcı bir yöntem olan boosting kullanılabilmektedir. Aynı zamanda hafızayı daha verimli kullanabilmekte, her iki algoritma sonuçları aynı olsa da biçim olarak bu algoritma ile daha düzgün karar ağacı elde edilmektedir. C5.0 algoritması daha küçük karar ağaçları oluşturabilir, verinin ayrıştırılması ve kirli verilerin iyileştirilmesinde daha etkilidir. Diğer yöntemlerden farklı olarak bilgi kazanımı (information gain) ölçütüyle ağacı büyütmektedir.

- SLIQ

- Mehta, Agrawal ve Rissanen uzmanlarının 1996 yılında geliştirdiği SLIQ algoritması, nitel ve nicel veri tiplerinde kullanılabilmektedir. Bu algoritma, dallanma kriterinin hesaplanmasında “Gini tekniği” nden faydalanmaktadır.

- SPRINT

- 1996 yılında Shafer, Agrawal ve Mehta tarafından geliştirilen SPRINT algoritması, entropiye dayanmaktadır. Bu algoritma, ağaç yapısında optimum dallanmayı sağlayabilmek için her bir değişkene ait verileri bir kez sıraya dizmekte ve ağaç yapısını bu şekilde oluşturmaktadır. Dallandırma kriteri olarak Gini indeksini kullanmaktadır.

- QUEST

- 1997 yılında Loh ve Shih uzmanları tarafından geliştirilen QUEST algoritması, dallanma sürecinde optimum bölünmeyi sağlayacak değişkene ve optimum bölünme sağlanacak noktanın karar verilmesine ayrı zamanlar ayırmaktadır. Bu yönüyle CART ve CHAID algoritmalarından ayrılmakta ve sınıflandırma modellerinde daha hızlı sonuç sağlamaktadır

- MARS

- 1990'ların başında istatistikçi Friedman tarafından geliştirilen Mars algoritması nonparametrik ve lineer olmayan bir yöntemdir. MARS yöntemi regresyon modellemesinde yeni bir yaklaşım olup, geleneksel yöntemlere göre son derece zor olan, çok boyutlu verilerin içinde gizlenmiş karmaşık veri yapısını, en uygun veri dönüşümlerini ve verilerin karşılıklı etkileşimlerini belirleyebilme avantajına sahip bir yaklaşımdır

Decision Tree

ITEM THAT YOU WANT TO USE

Does it have to be that exact item?

YES

NO

Conduct Fair Use evaluation

Get permissions and follow restrictions

Link to the source

Restricted access?

Used as comment / parody?

Transformative use?

Use a clip / portion?

Provide attribution if possible

Search library-licensed materials

Create your own

Find an open source option

Provide attribution and post in a non-reproducible format

Add a CC license + share

Search public domain materials

Search CC-licensed materials

Provide attribution if possible

Add CC license with appropriate attribution

Resources

Creative Commons <http://creativecommons.org/>

Creative Commons Image Search <http://search.creativecommons.org/>

Fair Use Evaluation Tool <http://librarycopyright.net/resources/fairuse/>

KARAR AĞACININ AVANTAJLARI

- Karar ağacı, herhangi bir matematiksel anlayış olmadan kolayca yorumlanabilir.
- Oluşturma süreci hızlıdır.
- Hem sayısal hem kategorik verileri işleyebilir.
- İstatistikler ile doğrulanabilir.



KARAR AĞACININ AVANTAJLARI

- Görseller ile desteklenebilir.
- Minimum bilgi ile oluşturulabilir.
- Çok çıktılı problemleri inceyebilir.



KARAR AĞACININ DEZAVANTAJLARI

- Veriyi iyi bir şekilde açıklayamayan aşırı karmaşık ağaçlar üretilebilir. Bu durumda dallanma takip edilemeyebilir.
- Ağaç eğitim verilerine daha sonra tam mantıksal sonuna kadar ağaç overfits büyümeye izin verilir. Bu durum kontrol edilemezse kalıtsal bir sorun haline gelebilir.
- Hem ağaç oluşturma karmaşıklığı hem de ağaç budama karmaşıklığı fazladır.

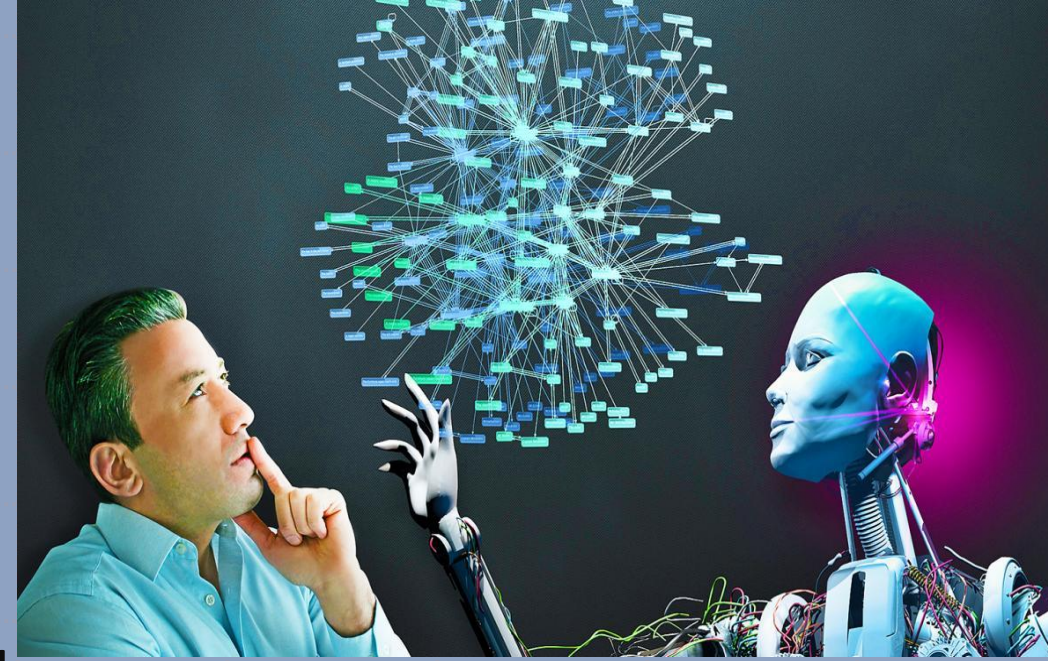


KARAR AĞACININ DEZAVANTAJLARI

- Sınıf sayısı fazla ve öğrenme kümesi örnekleri sayısı az olduğunda model oluşturma çok başarılı değildir.
- Sürekli nitelik değerlerini tahmin etmekte çok başarılı değildir.

KARAR AĞAÇLARININ YAPAY ZEKA İÇİN ÖNEMİ

- Makine öğrenmesi uzun bir süreç olduğu için bu anlamda karar ağaçları oluşturma süresi hızlı olduğundan dolayı yapay zekaya önemli bir katkı sağlar.
- Makine öğrenmesi matematiksel olarak ifade etmekte güçlük çektiğimiz fakat mantıksal ilişkinin var olduğuna emin olduğumuz durumlarda kullanılmaktadır, karar ağaçları kolayca yorumlanabildiği için yapay zekaya katkı sağlar.



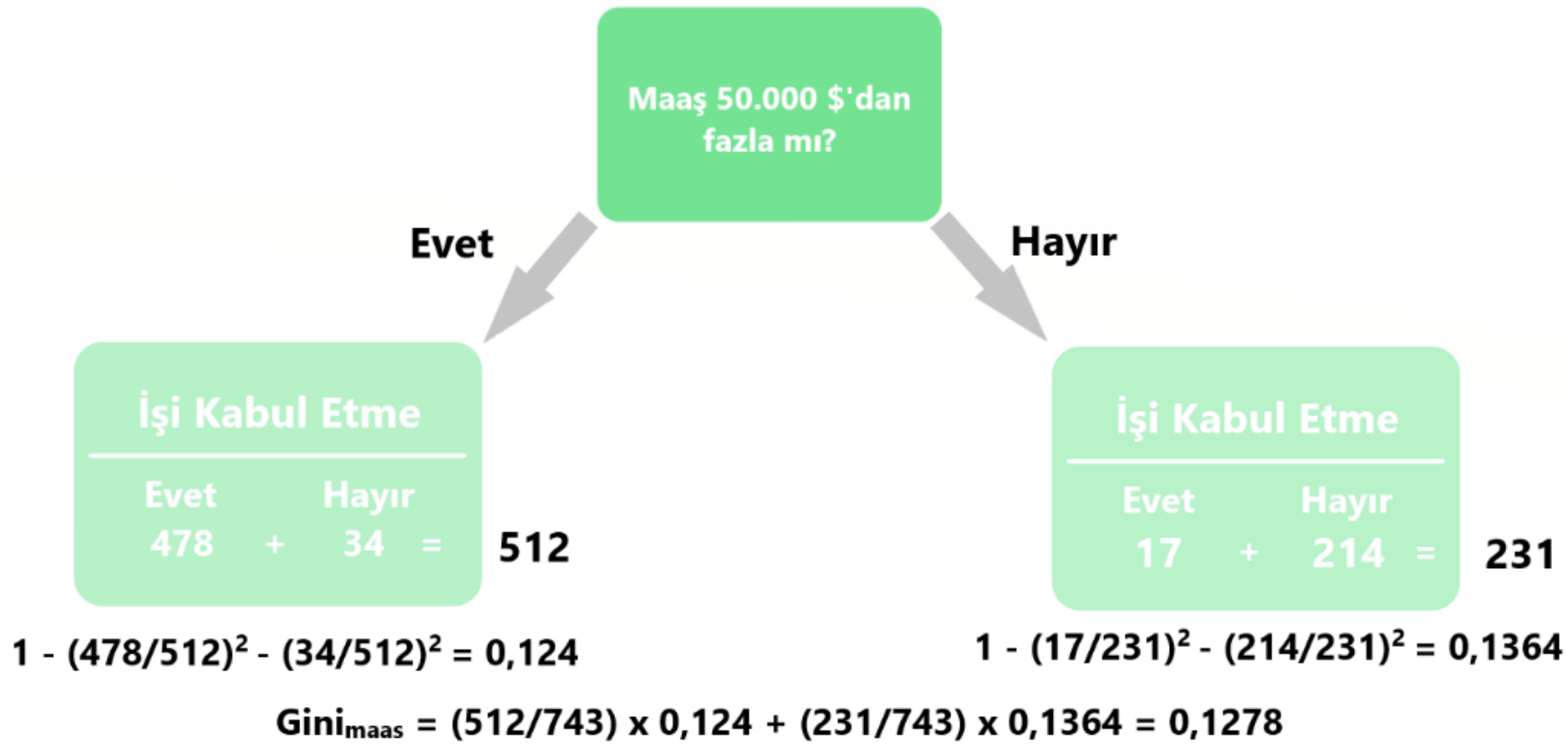
KARAR AĞAÇLARININ YAPAY ZEKA İÇİN ÖNEMİ

- Karar ağaçları, veri madenciliğinde tahminleyici ve tanımlayıcı niteliklere sahip olması, uygulamasının ve değerlendirilmesinin kolay oluşu, güvenilirlik düzeylerinin yüksek olması ve bilgisayar sistemlerindeki depolama ünitelerinde kolay entegre edilebilmeleri sebebiyle yapay zeka için önemlidir.

ÖRNEK : CART ALGORİTMASIYLA GİNİ DEĞERİNİ KULLANARAK KARAR AĞACI OLUŞTURMA

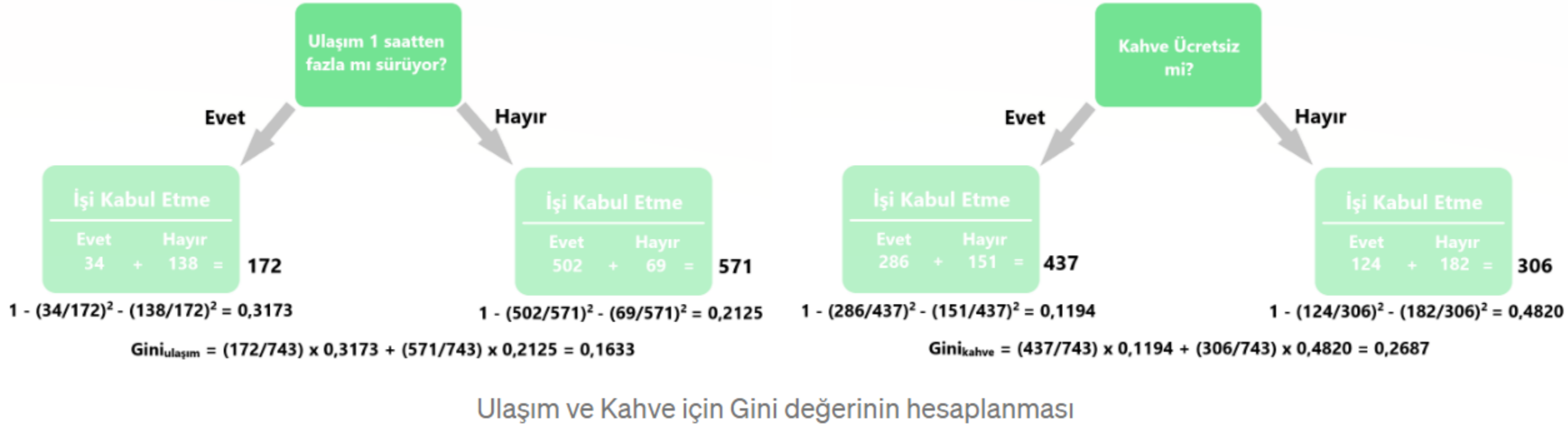


Kök (root) hücreyi bulabilmek için her düğüm için Gini değerini hesaplamamız gerekiyor.



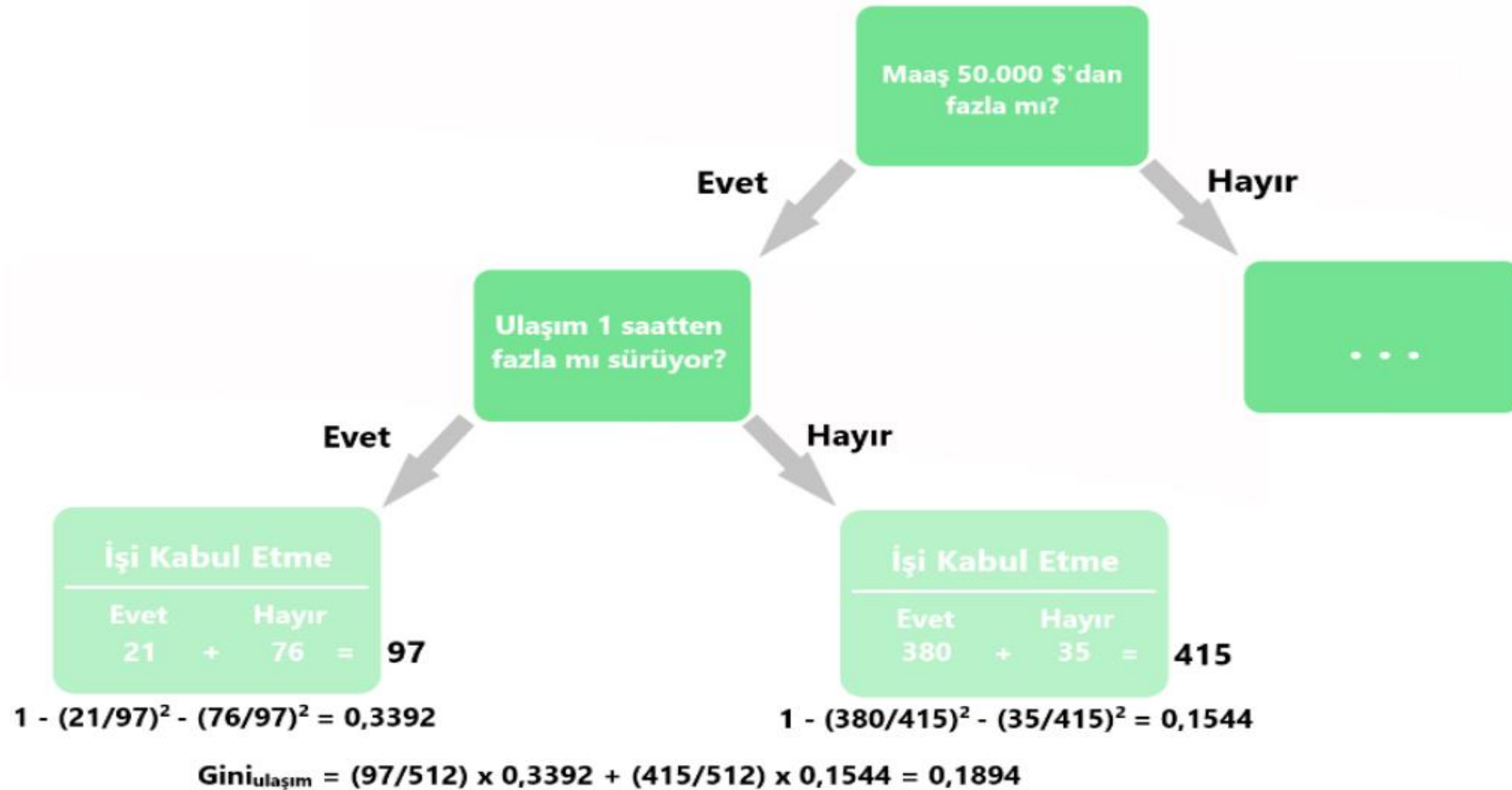
Maaş için Gini değerinin hesaplanması

- Maaş için inceleyelim. Öncelikle “Evet/Hayır” olarak ağacımızı ikiye ayırıyoruz. Maaş 50.000 \$’dan fazla olduğunda; 512 kişinin 478’i işi kabul ederken 34 kişi işi reddediyor. Ardından $1 - \text{maaş } 50.000 \$\text{’dan yüksek iken işin kabul edilme olasılığı} - \text{maaş } 50.000 \$\text{’dan yüksek iken işin reddedilme olasılığı}$ işlemini yaparak cevabı “Evet” olanların Gini değerini bulmuş oluyoruz. Aynı işlemi cevabı “Hayır” olanlar için tekrarlıyoruz. Son olarak $(\text{işi kabul edenlerin veri setindeki oranı} * \text{işi kabul edenlerin Gini değeri}) + (\text{işi reddedenlerin veri setindeki oranı} * \text{işi reddedenlerin Gini değeri})$ işlemini yapıyoruz ve maaş için 0,1278 değerini buluyoruz.



- Değerlere baktığımızda 0'a en yakın olanın Maaş olduğunu görüyoruz. Bu bize Maaş düğümü ile yapılan ayrımın veri setini diğerlerine göre daha iyi ayırttığını gösteriyor. Bu yüzden kök (root) hücre olarak Maaş düğümünü seçiyoruz.

- *Kök hücreden sonra hangisinin geleceğine de aynı şekilde karar veriyoruz.*



İç düğümlerin seçilmesi için Gini değerinin hesaplanması

ÖRNEK : ID3 ALGORİTMASIYLA KARAR AĞACI OLUŞTURMA

HAFTASONU	HAVA DURUMU	AİLE	PARA	KARAR
H1	Güneşli	Evet	Zengin	Sinema
H2	Güneşli	Hayır	Zengin	Tenis
H3	Rüzgarlı	Evet	Zengin	Sinema
H4	Yağmurlu	Evet	Fakir	Sinema
H5	Yağmurlu	Hayır	Zengin	Ev
H6	Yağmurlu	Evet	Fakir	Sinema
H7	Rüzgarlı	Hayır	Fakir	Sinema
H8	Rüzgarlı	Hayır	Zengin	Alışveriş
H9	Rüzgarlı	Evet	Zengin	Sinema
H10	Güneşli	Hayır	Zengin	Tenis

İlgili Durumlara Bağlı Olarak Yapılan Hafta Sonu Aktivitelerini İçeren Veri Kümesi

Etiket	Adet
Sinema	6
Tenis	2
Alışveriş	1
Ev	1

- Bir sınıf etiketinin gerçekleşme olasılığı (P_i) “ilgili etikete sahip örnek sayısı / veri kümesindeki toplam örnek sayısı” ile hesaplanır. Buna göre P_i olasılıklarının aşağıdaki gibi olduğunu söyleyebiliriz:

$$P(\text{Sinema}) = 6/10 \quad P(\text{Tenis}) = 2/10$$

$$P(\text{Alışveriş}) = 1/10 \quad P(\text{Ev}) = 1/10$$

- Olasılık hesabının tamamlanmasının ardından entropi formülü yardımı ile veri kümesinin entropi değerini hesaplayalım :

$$-\left(\frac{6}{10}\right) * \log_2(6/10) - \left(\frac{2}{10}\right) * \log_2(2/10) - \left(\frac{1}{10}\right) * \log_2(1/10) - \left(\frac{1}{10}\right) * \log_2(1/10) = 1,571$$

- Böylelikle veri kümesinin entropi değerinin 1,571 olduğunu elde ettik.

Hava Durumu					
Güneşli		Rüzgarlı		Yağmurlu	
Etiket	Adet	Etiket	Adet	Etiket	Adet
Sinema	1	Sinema	3	Sinema	2
Tenis	2	Alışveriş	1	Ev	1
Hava Durumu Etiket Sayıları					

Bir niteliğin entropisi hesaplanırken kenarlarının entropisinin ağırlıklı ortalaması alınır.

Tablodaki değerleri kullanarak “Hava Durumu” özneliğinin ağırlıklı entropisini hesaplayalım.

Öncelikle kenarların entropisinin hesaplanmasıyla başlayalım.

$$\text{Entropi(Güneşli)} = -(1/3)\log(1/3) - (2/3)\log(2/3) = 0,918$$

$$\text{Entropi(Rüzgarlı)} = -(3/4)\log(3/4) - (1/4)\log(1/4) = 0,811$$

$$\text{Entropi(Yağmurlu)} = -(2/3)\log(2/3) - (2/3)\log(2/3) = 0,918$$

Hava Durumu Niteliğinin Kenarları Entropisi

Kolayca hesaplanan kenar entropileri değerlerinden sonra, ikinci olarak yapılması gereken kazanç değerlerinin hesaplanması işlemidir.

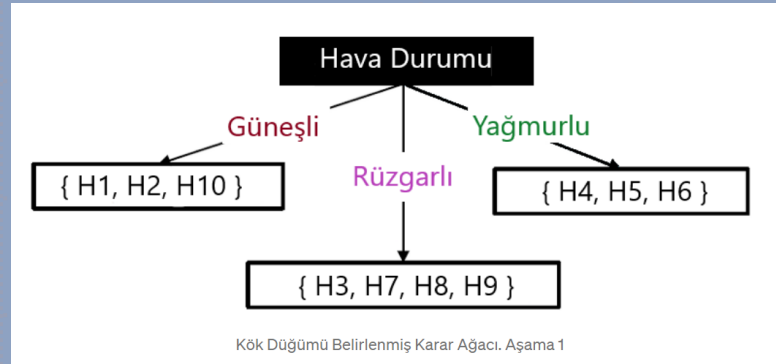
$$\text{Kazanç}(S, \text{Hava Durumu}) = 1,517 - [(3/10) \cdot 0,918 + (4/10) \cdot 0,811 + (3/10) \cdot 0,918] = 0,70$$

- Böylelikle “Hava Durumu” öznitelğine ait kazanç değeri hesaplanmış oldu. Bu işlem diğer tüm öznitelikler için tekrarlanır ve kazanç değeri en büyük olan öznitelik kök düğüm olarak seçilir. Diğer özniteliklerin kazanç değerleri özetle aşağıdaki gibi listelenebilir.

$$\text{Kazanç}(S, \text{Aile}) = 0,61$$

$$\text{Kazanç}(S, \text{Para}) = 0,2816$$

- Elde edilen kazanç değerlerinden görüleceği üzere, “Hava Durumu” düğümü, en çok kazanç sağlayan düğüm olarak “kök düğüm” olmaya hak kazanmıştır!



- Bu durumda kök düğümün her bir kenarı için hangi örneklerin geçerli olduğu belirlenir. Her kenar için sistemi ifade edecek veri kümesi bu belirlenen örneklerden oluşur. Bundan sonraki aşamalarda bu veri kümelerinin her biri ayrı bir sistem olarak ele alınır.

- Algoritmaya geri dönecek olursak bir sonraki aşamada tüm kenarlar için bir sonraki düğüm hesaplanır. Örnek olarak veri kümesindeki **“Hava Durumu”** özneliliğinin **“Güneşli”** kenarını hesaplayacak olursak;
- Veri kümesindeki ilgili örnekler alınır ve entropi hesabı bu veri kümesi üzerinden yapılır. Aşağıda bu örnekler gösterilmektedir.

HAFTASONU	AİLE	PARA	KARAR
H1	Evet	Zengin	Sinema
H2	Hayır	Zengin	Tenis
H10	Hayır	Zengin	Tenis

Hava Durumu özneliliği Güneşli olan kayıtlar

Etiket	Adet
Sinema	1
Tenis	2

$$\text{Entropi(Güneşli)} = -(1/3)\log(1/3) - (2/3)\log(2/3) = 0,918$$

$$\text{Entropi(Güneşli,Zengin)} = 0,918$$

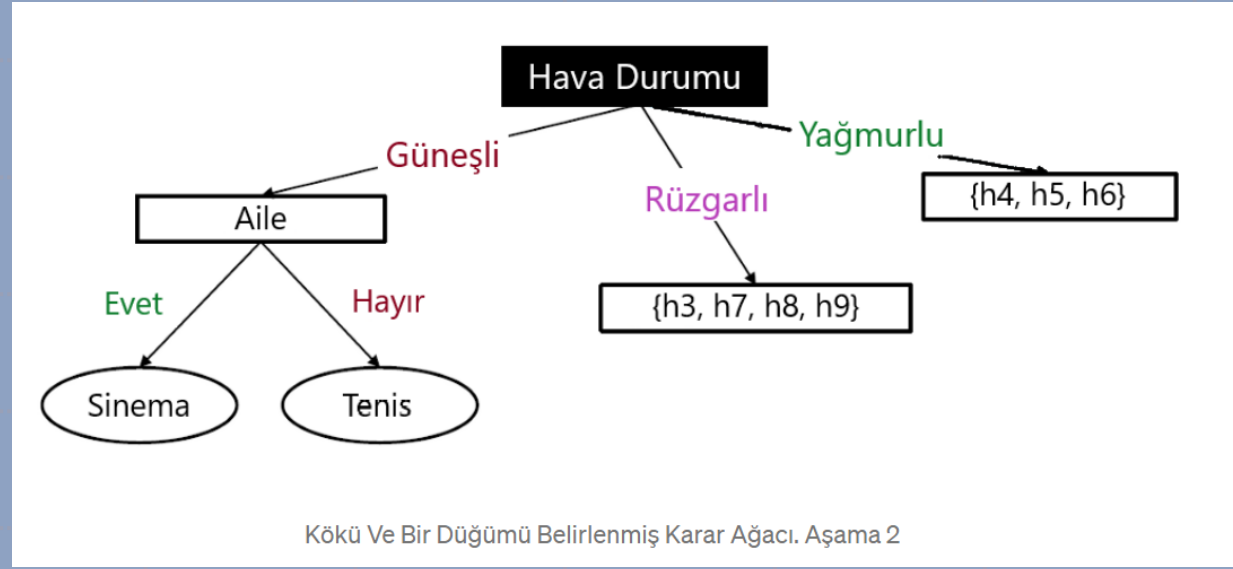
$$\text{Entropi(Güneşli,Fakir)} = 0$$

$$\text{Kazanç(Güneşli,Para)} = 0,918 - [(3/3)*0,918 + 0*0] = 0$$

- Kök düğümü bulmak için yaptığımız işlemleri ağacın diğer düğümlerini de belirlemek için yaptık. Kazanç değeri 0 çıkması bu niteliğin bu aşamada düğüm olarak belirlenemeyeceğini gösterir. Örnekteki para özneliliği yerine Aile özneliliğini ele alırsak : $\text{Kazanç(Güneşli,Aile)} = 0,918 - [(1/3)*0 + (2/3)*0] = 0,918$

Görüldüğü üzere Aile özneliliği Para özneliliğinden daha kazançlıdır bu nedenle düğüm olarak Aile özneliliği seçilmelidir.

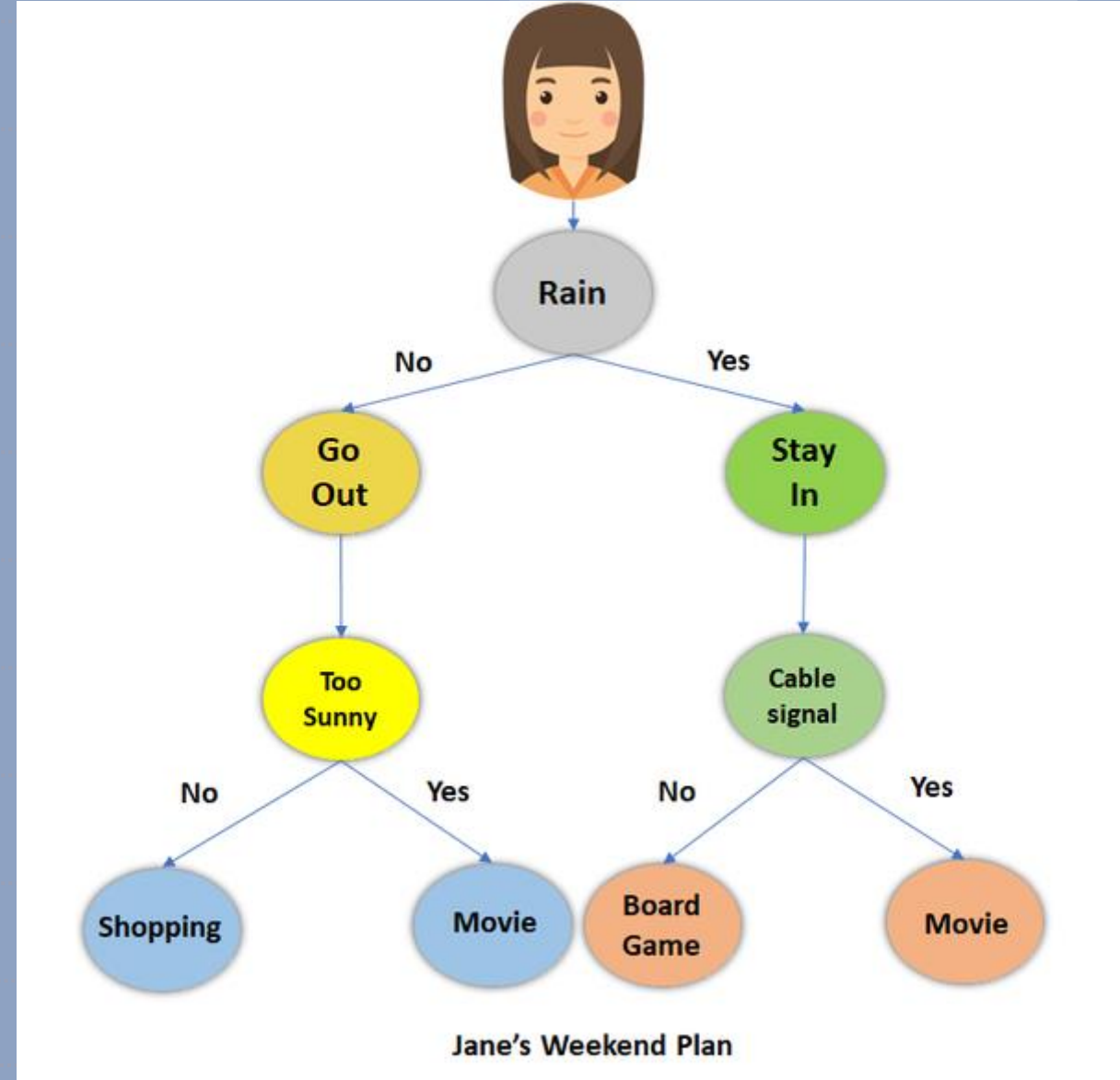
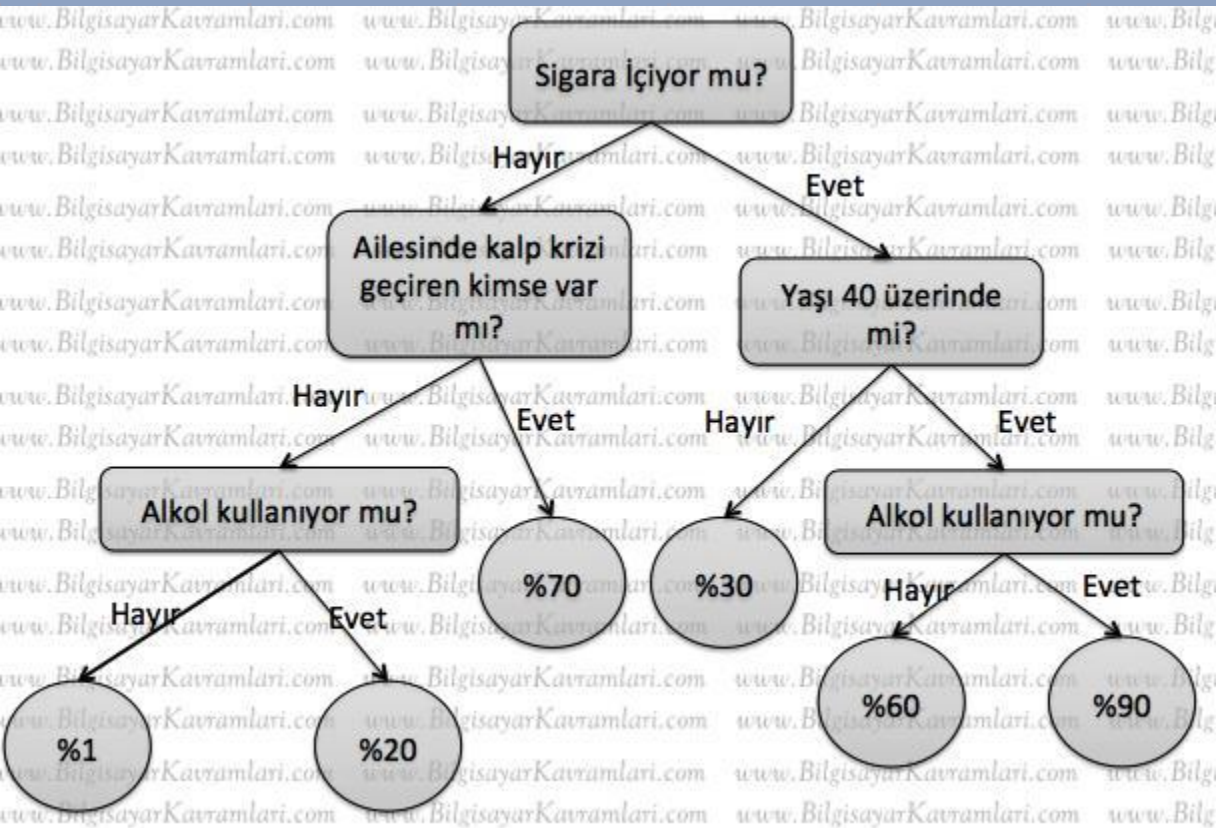
Önceki slaytlarda belirlenen işlemler gerçekleştirildikten sonra karar ağacının son hali aşağıda belirtildiği şekilde olur.



Düğümüleri belirlemek için yapılan bu işlemler veri kümesinin tüm kenarlarına uygulanarak kazanç hesabına uygun şekilde ağacın düğümleri oluşturulur ve karar ağacı tamamlanır. Ağacın yaprak kısmına yani en altına ulaşmak için entropi değerinin 0 olması gerekmektedir. Entropi değerinde 0'a ulaşıldığı takdirde karar ağacı tamamlanmış olur.

ÖRNEK KARAR AĞAÇLARI

Kalp krizi riskini hesaplayan karar ağacı



ÖRNEK VERİ SETİ

yaş	gelir	öğrenci	kredi durumu	Bilgisayar Alır?
≤30	yüksek	hayır	vasat	hayır
≤30	yüksek	hayır	mükemmel	hayır
31 ... 40	yüksek	hayır	vasat	evet
>40	orta	hayır	vasat	evet
>40	düşük	evet	vasat	evet
>40	düşük	evet	mükemmel	hayır
31 ... 40	düşük	evet	mükemmel	evet
≤30	orta	hayır	vasat	hayır
≤30	düşük	evet	vasat	evet
>40	orta	evet	vasat	evet
≤30	orta	evet	mükemmel	evet
31 ... 40	orta	hayır	mükemmel	evet
31 ... 40	yüksek	evet	vasat	evet
>40	orta	hayır	mükemmel	hayır

DEĞİŞKEN SEÇİMİ ÖLÇÜSÜ: BİLGİ KAZANIMI - HESAPLAMA

- P Sınıfı: Bilgisayar Alır? = “evet”
- N Sınıfı: Bilgisayar Alır? = “no”
- $I(p, n) = I(9, 5) = 0.940$
- Yaş için entropiyi hesaplayalım:

yaş	p_i	n_i	$I(p_i, n_i)$
≤ 30	2	3	0.971
31...40	4	0	0
> 40	3	2	0.971

yaş	gelir	öğrenci	kredi durumu	Bilgisayar Alır?
≤ 30	yüksek	hayır	vasat	hayır
≤ 30	yüksek	hayır	mükemmel	hayır
31...40	yüksek	hayır	vasat	evet
> 40	orta	hayır	vasat	evet
> 40	düşük	evet	vasat	evet
> 40	düşük	evet	mükemmel	hayır
31...40	düşük	evet	mükemmel	evet
≤ 30	orta	hayır	vasat	hayır
≤ 30	düşük	evet	vasat	evet
> 40	orta	evet	vasat	evet
≤ 30	orta	evet	mükemmel	evet
31...40	orta	hayır	mükemmel	evet
31...40	yüksek	evet	vasat	evet
> 40	orta	hayır	mükemmel	hayır

$$E(yaş) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694$$

$\frac{5}{14} I(2,3)$ 'ın manası, 14 örnekten, 2'si evet ve 3'ü de hayır olmak üzere toplam 5 “yaş ≤ 30 ” örneği vardır. Böylece

$$Kazanç(yaş) = I(p, n) - E(yaş) = 0.246$$

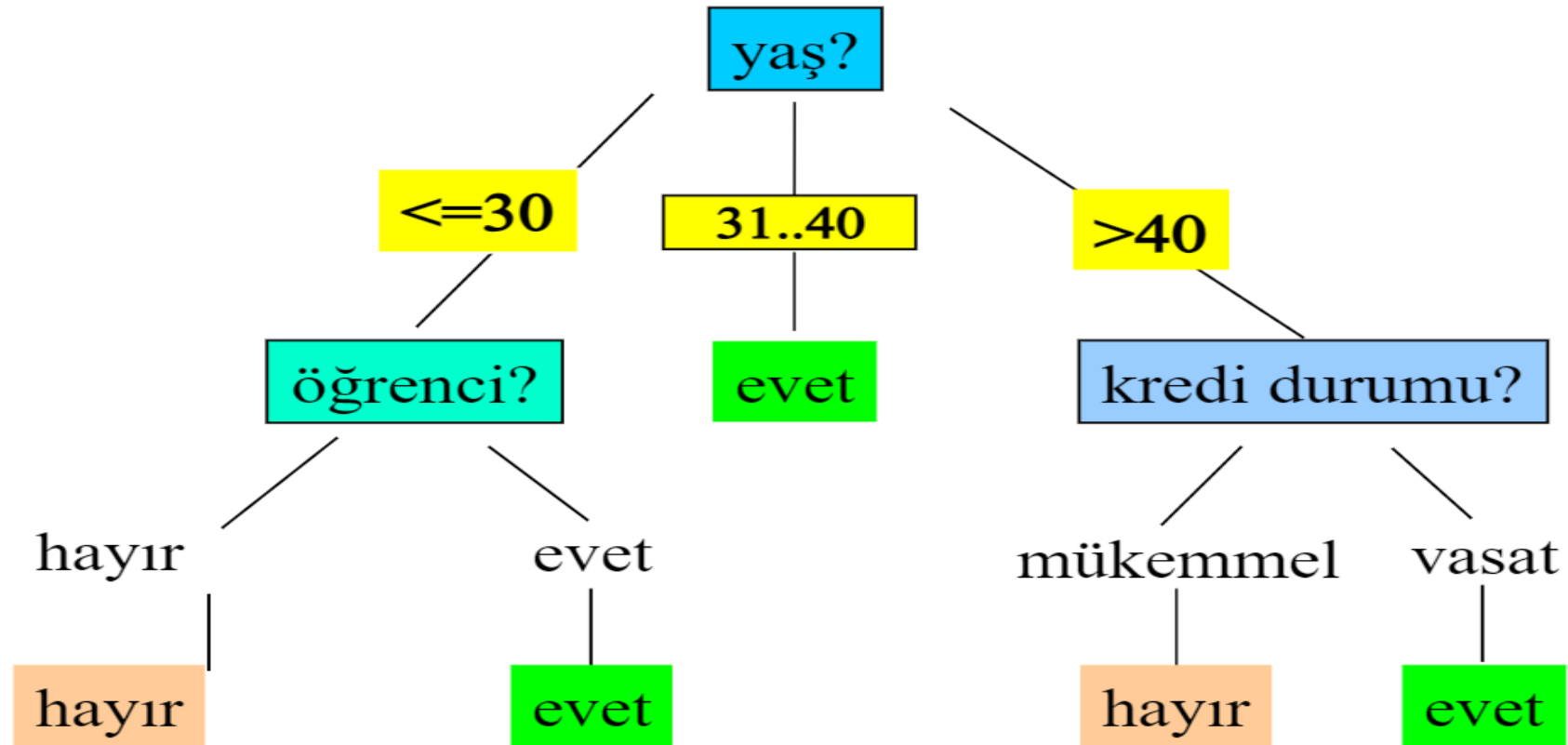
Buna benzer,

$$Kazanç(gelir) = 0.029$$

$$Kazanç(öğrenci) = 0.151$$

$$Kazanç(kredi durumu) = 0.048$$

ÖRNEK KARAR AĞACI



KAYNAKÇA

- ❖ http://www.tarmakbir.org/haberler/Kitap/10_3.4%20Yapay%20Zeka%20Y%C3%B6ntemleri.pdf
- ❖ <https://medium.com/deep-learning-turkiye/karar-a%C4%9Fa%C3%A7lar%C4%B1-makine-%C3%B6%C4%9Frenmesi-serisi-3-a03f3ff00ba5>
- ❖ <https://www.datascienceearth.com/karar-agaclarinda-algoritma-secimi/>
- ❖ <https://medium.com/deep-learning-turkiye/makine-%C3%B6%C4%9Frenmesi-algoritmalar%C4%B1-id3-algoritmas%C4%B1-71983b3e3b77>
- ❖ <https://medium.com/@k.ulgen90/makine-%C3%B6%C4%9Frenimi-b%C3%B6l%C3%BCm-5-karar-a%C4%9Fa%C3%A7lar%C4%B1-c90bd7593010>
- ❖ <https://ichi.pro/tr/karar-agaci-modeline-genel-bakis-42316283748679>
- ❖ https://erdincuzun.com/makine_ogrenmesi/decision-tree-karar-agaci-id3-algoritmasi-classification-siniflama/#:~:text=Karar%20a%C4%9Fa%C3%A7lar%C4%B1%20%E2%80%93%20s%C4%B1n%C4%B1flama%2C%20%C3%B6zellik%20ve,daha%20k%C3%BC%C3%A7%C3%BCk%20par%C3%A7alara%20b%C3%B6lerek%20geli%C5%9Ftirilir.