C4.5 Algoritması

Veri Madenciliği –Karar Ağacı Oluşturma

Murat TEZGİDER

C4.5 Algoritması

- ID3 algoritmasını geliştiren Quinlan'ın geliştirdiği C4.5 karar ağacı oluşturma algoritmasıdır.
- ID3 algoritmasında bazı eksiklikler ve sorunlar vardır.
- Bu sorunlar yine Quinlan'ın geliştirdiği C4.5 algoritmasıyla giderildi

C4.5 Algoritması

- C4.5 Algoritması ID3 algoritmasının bütün özelliklerini kendine miras alarak oluşturulmuş bir algoritmadır.
- ID3 için bahsedilen özelliklere yenileri eklenmiştir.

C4.5 Algoritmasının ID3 Algoritmasından Fazlaları

1. Bölünme-Dağılma Bilgisi (Split-Info)
2. Özelliklerin kayıp değerleriyle baş edilmesi
3. sayısal özellik değerlerinin hesaba katılması

1.Bölünme-Dallanma bilgisi (Split Information)-1

- Bir kategorik özelliğin olası değer çeşitliliği ne kadar yüksek olursa o özelliğin bilgi kazancı gereksiz bir şekilde yüksek çıkar ve bu durum ağacın doğruluğunu kötü bir şekilde etkiler.
- Bu tip özellikler işe yaramadıkları gibi bilgi kazancı yüksek özelliklerin de önüne geçip veride gizlenmiş kuralların çıkarılmasına engel teşkil ederler.

Örnek(Tablo2.1)

	Büyüklük	Renk	Biçim	Sonuç
1	Orta	Mavi	Tuğla	Evet
2	Küçük	Kırmızı	Kama	Hayır
3	Küçük	Kırmızı	Küre	Evet
4	Geniş	Kırmızı	Kama	Hayır
5	Geniş	Yeşil	Sütun	Evet
6	Geniş	Kırmızı	Sütun	Hayır
7	Geniş	Yeşil	küre	Evet

Bu maddenin büyüklük, renk ve şekil gibi özellikleri olsun ve 7 adet örnek olay olsun. Bu örnekler evet-hayır olarak ikili sınıflandırılmış olarak gösterilmiştir.

- Tablo2.1'deki 1'den 7'e kadar verilmiş olan etiket numaraları bir özelliğe karşılık gelsin. Bu aşamada bu özelliğin bilgi kazancı hesap edilsin;
- 1:evet, 2:hayır, 3:evet,4:hayır,5:evet,6:hayır ve 7:evet şeklinde her bir özellik değeri için bir tane sonuç elde edilecektir.
- Bu durumda 1 değeri için gereken bilgi kazancı

$$-(1 \times \log_2 1/1) - (0 \times \log_2 0/1) = 0$$

aslında tüm değerleri için gereken bilgi kazancı 0 çıkacaktır.

- Etiket özelliği için ortaya çıkan bilgi kazancı ise

$$0 \times (1/7) + 0 \times (1/7) + 0 \times (1/7) + 0 \times (1/7) + 0 \times (1/7) + 0 \times (1/7) + 0 \times (1/7) = 0$$

- Genel bilgi teorisi için bu sonucu mevcut bilgi gereksiniminden çıkarmak gerekecek. Bu durumda etiket özelliği için bilgi kazancı: 0.99 – 0 = 0.99 olur
- Bu sonuç diğer sonuçlar arasındaki en yüksek sonuçtur ve buna göre bir ağaç oluşturulduğunda Aşağıdaki ağaç yapısı oluşur.

- Bilgi kazancının yüksek çıkmasının sebebi özellik çeşitliliğinin yüksek olmasıdır.

- İşte bu tip gereksiz bilgilerle başa çıkabilmek için Quinlan bölünme bilgisi kavramı ile algoritmasını güncellemiştir.

- Bu algoritma değer çeşitliliği fazla olan özelliklerin bilgi kazancını azaltarak algoritmanın gereksiz bazı çıkarımlar yapmasını engellemektedir.

- Bu noktada bölünme bilgisi denilen yeni bir kavram ekleniyor bu algoritmaya. A bir özellik, Ai bu özelliğin değerleri, Ti Ai özelliğinin bu veride kaç kez tekrarlandığı ve T ise ele alınan olay sayısını temsil etsin. Bu durumda bölünme bilgisi aşağıdaki gibi hesaplanır.

$$-\sum_{i=1}^{n} \frac{Ti}{T} \times \log_2 \left(\frac{|Ti|}{|T|} \right)$$

- Beklenen bilgi gereksinimi:

$$E(A) = \sum_{i-1}^{n} \frac{pi + ni}{p + n} I(pi, ni)$$

- Bilgi Kazancı:

Bilgi kazancı(A)= I(p,n) – E(A)

- Bu bölünme bilgisi tüm özelliklerin bilgi kazanç formülüne bölen olarak eklenir ve bu kazanç oranı olarak ifade edilir. Bu durumda A özelliğinin kazanç oranı :

kazanç_oranı=bilgi_kazancı(A) / bölünme_bilgisi(A)

- Yeni algoritma ile oluşturulan değerler:

Özellik ismi	Bilgi kazancı	Bölünme bilgisi	Kazanç oranı
Etiket	0.99	2.80	0.35
Büyüklük	0.13	1.38	0.09
Renk	0.52	1.38	0.37
Biçim	0.7	2.2	0.31

2.Sayısal Özellikler

- Veri kümesinde iki tip veri vardır; Nominal (kategorik) ve sayısal. Nominal daha önceki bölümlerde kullanılan veri tipleridir.
- Örnek olarak renk özelliği mavi, kırmızı ve sarı ile, büyüklük özelliği büyük, orta ve küçük ile ifade edilir.
- ID3 algoritması daha önce sadece nominal değerlere sahip veri tipleri ile işlemler yapabilirdi. Ancak bu algoritmanın varisi olan C4.5 algoritması ise bu tip veriler için de bir yöntem geliştirmiştir.

2.Sayısal Özellikler

- İlk bakışta sayısal özelliklerle uğraşmak ve onların bilgi kazancını hesaplamak oldukça zor gelebilir. Ancak bu iş o kadar da zor değildir. Yapılması gereken iş sadece bu özelliğin sayısal değerleri arasında uygun eşik değerini bulmaktır.
- Bu eşik değeri bulunduktan sonra ikili bir bölünme ile veri kümesi bölünebilir; Bu eşik değerinden büyük veriler ile bu eşik değerinden küçük veriler
- Bu anlamda algoritma çok sade bir şekilde açıklanabilir.

Eşik Değerini Belirleme

- Öncelikle tüm sayısal değerler küçükten büyüğe sıralanır.
- Bu sıra {v1,v2,...,vm} ile ifade edilsin.
- Bu durumda seçilen eşik değeri vi ve vi+1 arasında olursa

{v1, v2,...,vi} ile {vi+1, vi+2,...,vm} gibi iki grup ortaya çıkar.

- Buradan da görülüyor ki m-1 adet eşik değeri seçilebilir.
- Bu seçim işlemi için olası bütün eşik değerleri

$$\frac{v_i + v_{i+1}}{2}$$

- Formülü ile hesaplanır.

- Bu yapıyla sanki söz konusu özellik büyük-küçük değerleri olan

nominal bir özelliktir. Bu anlayışla nominal değerlere uygulanan bilgi oranı formülü tüm eşik değerleri için uygulanır ve bilgi kazanımı en iyi olan eşik değeri söz konusu özelliğin eşiği olarak kabullenilir.

- Eğer en iyi eşik değeri e ise ve söz konusu özelliğin
- sayısal değerleri {v1, v2,...,vn} kümesi ile ifade ediliyorsa, bu kümedeki
- vi< e koşulunu sağlayan elemanlar küçük kategorisine ve
- vi> e koşulunu sağlayan elamanlar büyük kategorisine dahil edilir.

3.Kayıp Veriler

- Bir önceki bölümde gizli bir kabullenme vardır; söz konusu ID3 algoritması eksik olmayan bir veri yığınını dikkate alır. Fakat veride bazı bilgilerin bulunmaması tüm algoritmanın çalışmasını engeller ve yanlış çıkarımlara yol açar.
- Eğer gerçek uygulamalar hedef alınırsa, böyle bir sorunla her zaman karşılanacaktır ve bu durum her zaman için kaçınılmazdır. Veri, çeşitli sebeplerden dolayı eksik olabilir.
- Verinin bazı özeliklerinin toplanması özgün koşullardan dolayı zor olabilir, veri formatına uymayan bir bilgi alınmış olabilir, veri tabanına aktarırken eksik girilmiş olabilir vb. bir çok sebepten dolayı veri bütünlüklü olmayabilir.

3.Kayıp Veriler

- Bu aşamada algoritmanın önünde üç problem durmaktadır.

a) Bilgi kazancı ve bilgi oranı değerleri olmayan özellikler için nasıl hesaplanır?

- b) KA'nı oluştururken alt ağaç yaratma işlemi sırasında özellik değeri olmayan
- satırlar hangi alt ağaca eklenecektir?
- c) KA oluşturduktan sonra, kayıp değerleri olan yeni bir olay sınıflandırılırken nasıl bir test yapılacaktır?

Kayıp Veriler Bilgi Kazancı

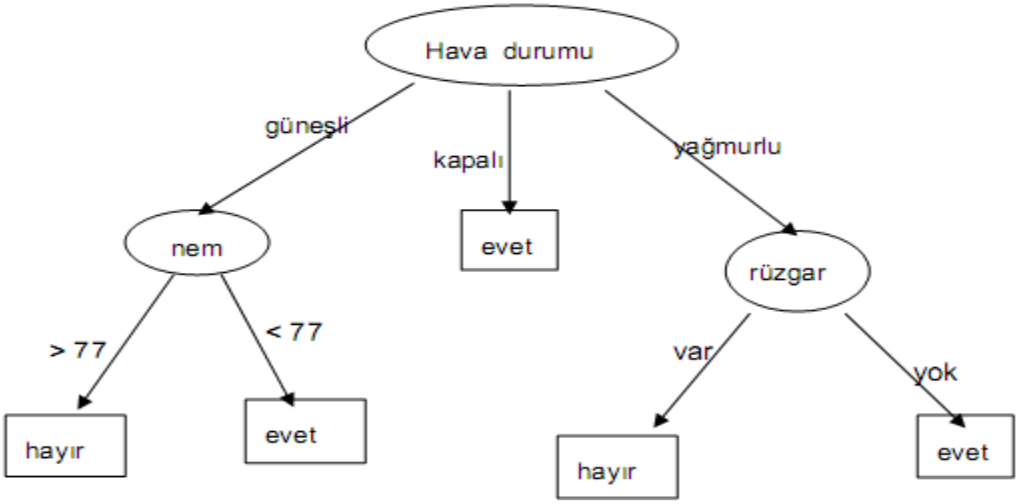
- Bir önceki bölümde bilgi kazancı hesap etme formülleri eksik olmayan verilere göre yapılmıştır. Şimdi ise bu eksik olan verilerde dikkate alınarak formüller güncellenecektir.
- T çalışılan veri kümesi olsun ve genel bilgi kazancı bilgi(T) olsun. X bu kümenin herhangi bir özelliği olsun ve X özelliğinin bilgi kazancı ise bilgi X(T) olsun. Bilgi(T) eskisi gibi hesap edilir. Ancak bilgi X(T)'yi hesap edilirken olmayan veriler bu kümeden çıkarılır. Olay sayısı n ile ifade edilirse ve bilinmeyen veriler b ile ifade edilirse X özelliğin n-b adet eksik olmayan verileriyle sanki hiçbir veri eksik değilmiş gibi klasik formül uygulanır. Ardından eksik olmayan değerlerin toplam değerlere oranı
- $F = (n-b) / n$ Bu durumda Bilgi : $F \bullet (bilgi(T) - bilgi_X(T))$

- Bu noktada bölünme bilgisini de hesaplara eklemek gerekecektir. Bölünme bilgisi hesabı sırasında olmayan verilere sanki X özelliğinin bir değeri gibi muamele edilir. Yani k adet çeşitliliğe sahip X özelliğinin çeşitliliği k+1 olur ve buna göre bölünme bilgisi standart formül ile hesap edilir.

Tablo 2.3 - Hava durumunun tenis oynamaya müsait olup olmadığına ilişkin veri kümesi

	Hava durumu	Sıcaklık (°F)	Nem(%)	Rüzgar	Sonuç
1	Güneşli	75	70	Var	Evet
2	Güneşli	80	90	Var	Hayır
3	Güneşli	85	85	Yok	Hayır
4	Güneşli	72	95	Yok	Hayır
5	Güneşli	69	70	Yok	Evet
6	Kapalı	72	90	Var	Evet
7	Kapalı	83	78	Yok	Evet
8	Kapalı	64	65	Var	Evet
9	Kapalı	81	75	Yok	Evet
10	Yağmurlu	71	80	Var	Hayır
11	Yağmurlu	65	70	Var	Hayır
12	Yağmurlu	75	80	Yok	Evet
13	Yağmurlu	68	80	Yok	Evet
14	Yağmurlu	70	96	Yok	Evet

- C4.5 algoritmasının böyle bir veri yığınından çıkardığı karar ağacı Şekil2.10'daki gibidir. Karar ağacının ilk sorusu hava durumu oldu ve 5 yapraklı ya da 5 kurallı bir karar ağacı yaratıldı.



Şekil 2.10 - Tablo2.3'teki veri kümesinin C4.5 ile yaratılması.

Bu verideki özelliklerin bilgi kazançları tablodaki gibidir. Bu veri kümesindeki kazanç oranları Tablo2.4 'te verilmiştir

Tablo 2.4 -Tablo2.3'teki verilerin bilgi kazanç oranları

Özellik	**Bilgi kazanç oranı**
Hava durumu	0.156
Sıcaklık	0.01
Nem oranı	0.151
Rüzgar	0.04

Eğer ki veri yığınımızda bazı özelliklerin değerleri olmasaydı nasıl bir karar ağacı ortaya çıkacaktı.

Tablo 2.5 - Tablo2.3'teki 6. satır

6	Kapalı	72	90	Var	Evet

Tablo 2.6 - 6.satırın hava durumu özelliğinin kayıp olması

6	?	72	90	Var	Evet

- Tablo 2.6’da ifade edildiği gibi “?” ile ifade edilsin. 14 satırlık bu veri tabanının artık başlangıç için değerleri bilinen ve sağlam 13 satırı ele alınacaktır. Bu durumda

Tablo 2.7 - Tablo2.5'in 6. satırındaki hava durumu bilgisinin kayıp olduğunda ortaya çıkan istatistik görüntüsü

Hava durumu	Evet	Hayır	Total
Güneşli	2	3	5
Kapalı	3	0	3
Yağmurlu	3	2	5
Toplam	8	5	13

- Tablo2.7'deki gibi bir tablo elde edilir. Hava durumu özelliğinin bilgi kazançları öncellikle sanki 6. satır hiç yokmuş gibi hesap edilir.

$$\text{Bilgi}(T) = -\frac{8}{13} \times \log_2\left(\frac{8}{13}\right) - \frac{5}{13} \times \log_2\left(\frac{5}{13}\right) = 0.961$$

$$\text{Bilgi}_X(T) = \frac{5}{13} \times \left(-\frac{2}{5} \times \log_2\left(\frac{2}{5}\right) - \frac{3}{5} \times \log_2\left(\frac{3}{5}\right)\right)$$

$$+ \frac{3}{13} \times \left(-\frac{2}{5} \times \log_2\left(\frac{2}{5}\right) - \frac{0}{3} \times \log_2\left(\frac{0}{3}\right)\right)$$

$$+\frac{5}{13} \times \left(-\frac{3}{5} \times \log_2\left(\frac{3}{5}\right) - \frac{2}{5} \times \log_2\left(\frac{2}{5}\right)\right) = 0.747$$

$$\text{Bilgi kazancı} = \frac{13}{14} \times (0.961 - 0.747) = 0.199$$

- Bu sonuç daha önce hesap edilen 0.246 ‘dan küçüktür. Çünkü veri yığınındaki verilerin bazıları bilinmiyor. Bölünme bilgisinin de hesap edilmesi gerekecek. Bölünme bilgisini hesap ederken eskiye göre yeni bir kategori oluşmuştur. Bu kategori “?”dir, yani kayıp veri kategorisidir. Bu değer güneşli, yağmurlu gibi değerlerin yanında kayıp değer olarak yer alacak.
- Bu durumda

$-5/14 \times \log_2 (5/14)$ → güneşli değeri için

$-3/14 \times \log_2 (3/14)$ → kapalı değeri için

$-5/14 \times \log_2 (5/14)$ → yağmurlu değeri için

$-1/14 \times \log_2 (1/14)$ → ? değeri için (formüldeki 1 bilinmeyen sayısını temsil ediyor)

- ile hesap edildiğinde daha önce 1.577 sonucunu veren bölünme bilgisi değeri 1.809 gibi bir değer alır.

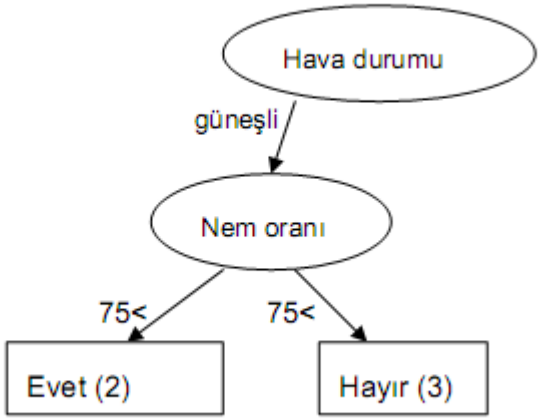
- 14 olay bu testler ile sınıflandırıldığı zaman bu olayların
- 13'ü hiç problemsiz ağaca yerleşirler. Ancak söz konusu 6.satır bu ağaçtaki yerini nasıl alacak ve hangi alt ağaca gönderilecek ? Bu aşamada bilinmeyen içerikli satır tüm alt ağaçlara ya da kümelere belli bir ağırlık bilgisiyle gönderilir. Çünkü tüm bu alt ağaçlara ait olma ihtimali vardır.
- Söz konusu satır hava durumu = güneşli, kapalı, yağmurlu 3 alt kümeye sırasıyla 5/13, 3/13, 5/13 ağırlık bilgileriyle gönderilir.

Birinci alt ağaca gönderilme durumu canlandırılırsa Tablo2.8 elde edilir.

Tablo2.8 - Hava durumu=güneşli alt ağacının kapsadığı kümenin ağırlık bilgileriyle gösterilmesi

	Hava durumu	Sıcaklık (°F)	Nem(%)	Rüzgar	Sonuç	Ağırlık
1	Güneşli	75	70	Var	Evet	1
2	Güneşli	80	90	Var	Hayır	1
3	Güneşli	85	85	Yok	Hayır	1
4	Güneşli	72	95	Yok	Hayır	1
5	Güneşli	69	70	Yok	Evet	1
6	?	72	90	Var	Evet	5/13

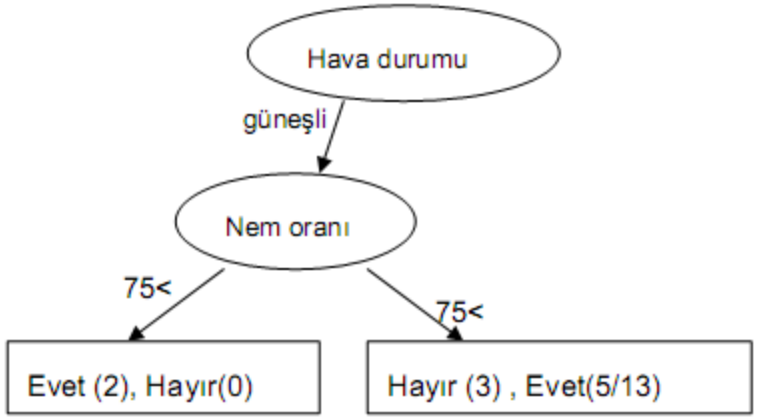
- Görüldüğü gibi 6. satır dışındaki bütün satırların ağırlıkları 1. Çünkü bu satırların bu alt ağaca ait olma olasılığı 1 olur. Bütün değerleri bilinen ilk ağaçta hava durumu = güneşli alt ağacı Şekil 2.11'deki gibidir.



Şekil 2.11 - kayıp veri göz önüne alınmadan oluşturulan alt ağaç.

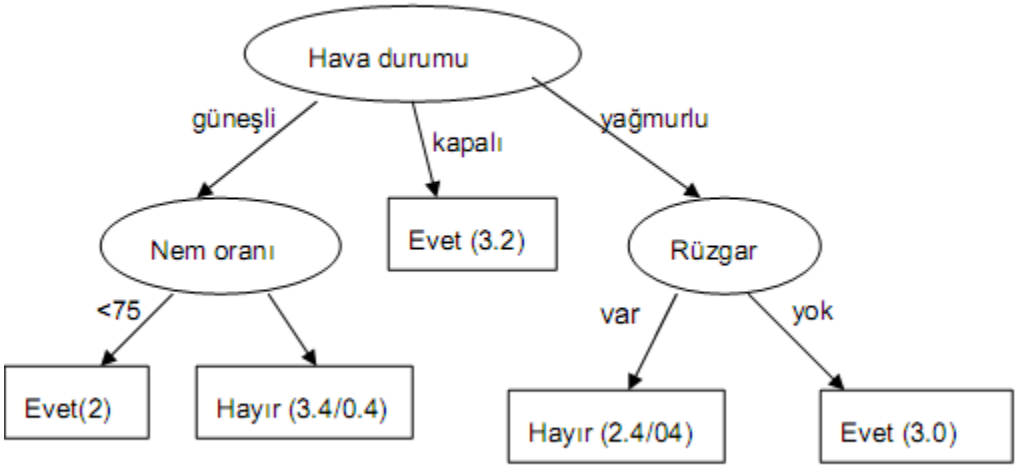
- Bu tablodaki parantez içindeki rakamlar o yaprağı oluşturan olay sayısını ifade ediyor.

6. satırının hava durumu “?” olduğunda bu alt ağaç şekil 2.12'deki



Şekil 2.12 - Kayıp veri göz önüne alındığında oluşan alt ağaç.

- gibi ifade edilir. Bu alt ağaçta artık iki sınıf hakkında da bilgi bulunmaktadır. Bu sınıf bilgileri olasılık olarak ifade edilir. Ancak sonuç olarak olasılığı en yüksek sınıfla yaprak etiketlenir. Tüm alt ağaçlar bu şekilde oluşturulduğunda Şekil2.13’teki ağaç elde edilir.



Şekil 2.13 - Hata içeriği ile ağacın gösterilmesi

Daha fazla bilgi için;

- Kaynak: **Yüksek Lisans Tezi - Tümevarim Öğrenme Tekniklerinden C4.5'in Incelenmesi (Savaş Yildirim)**

- Dinlediğiniz İçin Teşekkürler...