BAKIRÇAY

ÜNİVERSİTESİ

BİL365 - Makine Öğrenmesi

Dr. Öğretim Üyesi Murat UÇAR

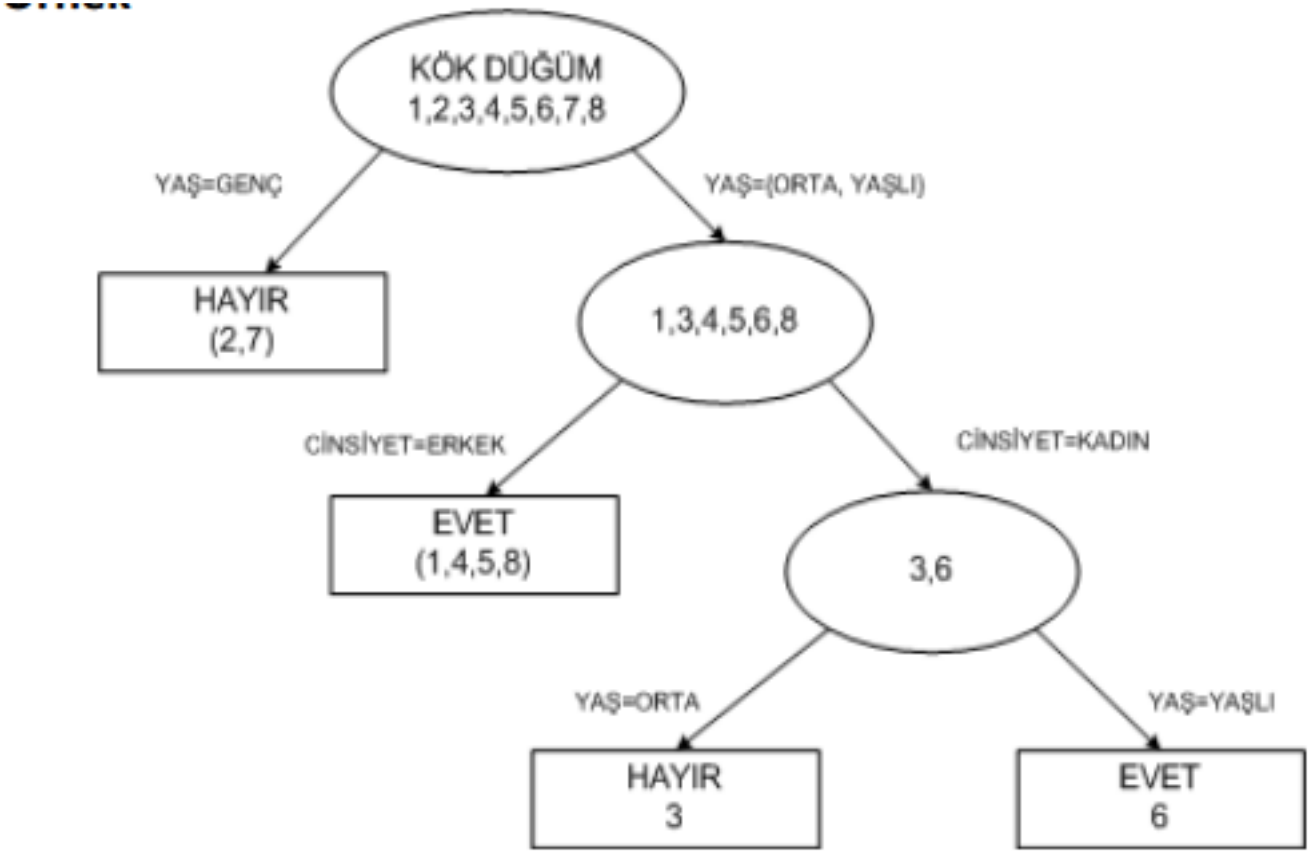
Sınıflandırma ve Regresyon Ağaçları (CART)

- Bu yöntem 1984'te Breiman tarafından ortaya atılmıştır. CART karar ağacı, herbir karar düğümünden itibaren ağacın iki dala ayrılması ilkesine dayanır. Yani bu tür karar ağaçlarında ikili dallanmalar söz konusudur.

- CART algoritmasında bir düğümde belirli bir kriter uygulanarak bölünme işlemi gerçekleştirilir. Bunun için önce tüm niteliklerin var olduğu değerler gözönüne alınır ve tüm eşleşmelerden sonra iki bölünme elde edilir. Bu bölünmeler üzerinde seçme işlemi uygulanır. Bu kapsamdaki iki algoritma bulunmaktadır.

  - Twoing Algoritması
  - Gini Algoritması

Örnek:



Twoing Algoritması

- Özniteliklerin içerdiği değerler dikkate alınarak eğitim kümesi iki ayrı dala ayrılır. Bunlara *aday bölünme* adı verilir. Bir t düğümünde "sağ" ve "sol" olmak üzere iki adet ayrı dal bulunur. Bu bölümlenen kümeler t_{sol} ve $t_{sağ}$ biçimindedir.
- Aday bölünmelerin her biri için P_{sol} ve $P(j|t_{sol})$ olasılıkları hesaplanır. $P(j|t_{sol})$ ifadesi bir j sınıf değerinin sol taraftaki bölünmede olma olasılığını verir.

$$P_{sol} = \frac{t_{sol}\,daki\ nitelik\ değerinin\ ilgili\ nitelik\ sütunundaki\ tekrar\ sayısı}{Eğitim\ kümesindeki\ kayıtların\ sayısı}$$

$$P(j|t_{sol}) = \frac{t_{sol}\,daki\ kayıtların\ j\ sınıfları\ sayısı}{t_{sol}\,daki\ herbir\ nitelik\ değerinin\ ilgili\ nitelik\ sütunundaki\ tekrar\ sayısı}$$

BAKIRÇAY ÜNİVERSİTESİ

Twoing Algoritması

- Aday bölünmelerin her biri için $P_{sağ}$ ve $P(j|t_{sağ})$ olasılıkları hesaplanır. $P(j|t_{sağ})$ ifadesi bir j sınıf değerinin sağ taraftaki bölünmede olma olasılığını verir.

$$P_{sağ} = \frac{t_{sağ}\,daki\ nitelik\ değerinin\ ilgili\ nitelik\ sütunundaki\ tekrar\ sayısı}{Eğitim\ kümesindeki\ kayıtların\ sayısı}$$

$$P(j|t_{sağ}) = \frac{t_{sağ}\,daki\ kayıtların\ j\ sınıfları\ sayısı}{t_{sağ}\,daki\ herbir\ nitelik\ değerinin\ ilgili\ nitelik\ sütunundaki\ tekrar\ sayısı}$$

- Aday bölünmenin uygunluk ölçüsü aşağıdaki şekilde hesaplanır.

$$\Phi(s|t) = 2P_{sol}P_{sağ}\sum_{j=1}^{n}|P(j|t_{sol}) - P(j|t_{sağ})$$

- Aday bölünme değerlerinden en büyük olanı seçilir.

Örnek

- Tabloda çalışanların maaş, deneyim, görev niteliklerine göre hedef niteliği olan memnun olma durumlarına ait 11 gözlem verilmiştir. Twoing algoritmasını kullanarak karar ağacını oluşturunuz.

PERSONEL	MAAŞ	DENEYİM	GÖREV	MEMNUN
1	NORMAL	ORTA	UZMAN	EVET
2	YÜKSEK	YOK	UZMAN	EVET
3	DÜŞÜK	YOK	YÖNETİCİ	EVET
4	YÜKSEK	ORTA	YÖNETİCİ	EVET
5	DÜŞÜK	ORTA	YÖNETİCİ	EVET
6	YÜKSEK	İYİ	YÖNETİCİ	EVET
7	DÜŞÜK	İYİ	YÖNETİCİ	EVET
8	YÜKSEK	ORTA	UZMAN	HAYIR
9	DÜŞÜK	ORTA	UZMAN	HAYIR
10	YÜKSEK	İYİ	UZMAN	HAYIR
11	DÜŞÜK	İYİ	UZMAN	HAYIR

Örnek

- Aday bölünmeler aşağıdaki gibidir.

BÖLÜNME	SOL	SAĞ
1	MAAŞ = NORMAL	MAAŞ = {DÜŞÜK, YÜKSEK}
2	MAAŞ = YÜKSEK	MAAŞ = {DÜŞÜK, NORMAL}
3	MAAŞ = DÜŞÜK	MAAŞ = {NORMAL, YÜKSEK}
4	DENEYİM = YOK	DENEYİM = {ORTA, İYİ}
5	DENEYİM = ORTA	DENEYİM = {YOK, İYİ}
6	DENEYİM = İYİ	DENEYİM = {YOK, ORTA}
7	GÖREV = UZMAN	GÖREV = YÖNETİCİ
8	GÖREV = YÖNETİCİ	GÖRE = UZMAN

Örnek (Sınıf Değerlerinin Olma Olasılığı)

- MAAŞ = NORMAL için

$$P_{sol} = \frac{|B_{sol}|}{|T|} = \frac{1}{11} = 0{,}09$$

$$P_{(EVET|t_{sol})} = \frac{|\text{Tsinif}_{EVET}|}{|B_{sol}|} = \frac{1}{1} = 1$$

$$P_{(HAYIR|t_{sol})} = \frac{|\text{Tsinif}_{HAYIR}|}{|B_{sol}|} = \frac{0}{1} = 0$$

BÖLÜNME	$\|B_{sol}\|$	P_{Sol}	$\|sinif_{EVET}\|$	$\|sinif_{HAYIR}\|$	$P(EVET\|t_{Sol})$	$P(HAYIR\|t_{Sol})$
1	1	0,09	1	0	1	0
2	5	0,45	3	2	0,6	0,4
3	5	0,45	3	2	0,6	0,4
4	2	0,18	2	0	1	0
5	5	0,45	3	2	0,6	0,4
6	4	0,36	2	2	0,5	0,5
7	6	0,55	2	4	0,33	0,67
8	5	0,45	5	0	1	0

- MAAŞ = {DÜŞÜK, YÜKSEK} için

$$P_{sag} = \frac{|B_{sag}|}{|T|} = \frac{10}{11} = 0{,}91$$

$$P_{(EVET|t_{sag})} = \frac{|\text{Tsinif}_{EVET}|}{|B_{sag}|} = \frac{6}{10} = 0{,}6$$

$$P_{(HAYIR|t_{sag})} = \frac{|\text{Tsinif}_{HAYIR}|}{|B_{sag}|} = \frac{4}{10} = 0{,}4$$

BÖLÜNME	$\|B_{sag}\|$	P_{sag}	$\|sinif_{EVET}\|$	$\|sinif_{HAYIR}\|$	$P(EVET\|t_{Sag})$	$P(HAYIR\|t_{Sag})$
1	10	0,91	6	4	0,6	0,4
2	6	0,55	4	2	0,67	0,33
3	6	0,55	4	2	0,67	0,33
4	9	0,82	5	4	0,56	0,44
5	6	0,55	4	2	0,67	0,33
6	7	0,64	5	2	0,71	0,29
7	5	0,45	5	0	1	0
8	6	0,55	2	4	0,33	0,67

Örnek

$$\Phi(s|t) = 2Pso_{lPsağ} \sum_{j=1}^{n} |P(j|t_{sol}) - P(j|t_{sağ})|$$

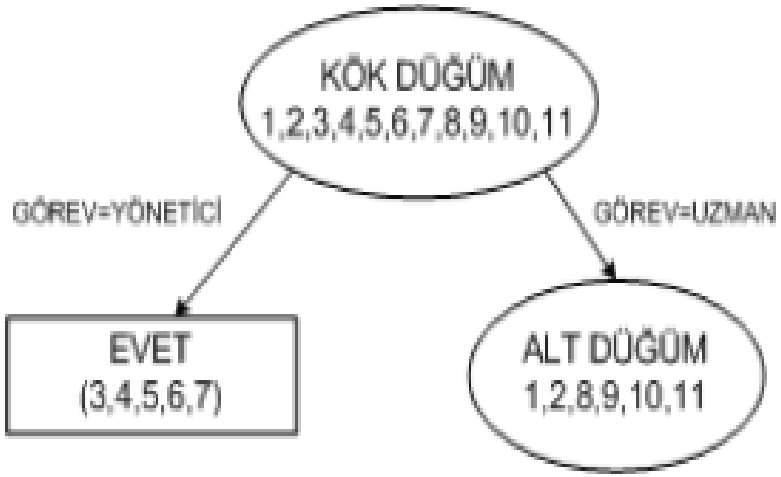
1. Aday Bölünme için hesaplama.

$$\Phi(s|t) = 2(0.09)(0.91)[|1.00 - 0.60| + |0 - 0.40|] = 0.131$$

BÖLÜNME	P_{sol}	$P_{sağ}$	$P(Evet|t_{sol})$	$P(Evet|t_{sağ})$	$P(Hayır|t_{sağ})$	$P(Hayır|t_{sol})$	Uygunluk Ölçüsü
1	0.09	0.91	1.00	0.60	0.40	0.00	0.131
2	0.45	0.55	0.60	0.67	0.33	0.40	0.069
3	0.45	0.55	0.60	0.67	0.33	0.40	0.069
4	0.18	0.82	1.00	0.56	0.44	0.00	0.260
5	0.45	0.55	0.60	0.67	0.33	0.40	0.069
6	0.36	0.64	0.50	0.71	0.29	0.50	0.194
7	0.55	0.45	0.33	1.00	0.00	0.67	0.663
8	0.45	0.55	1.00	0.33	0.67	0.00	0.663

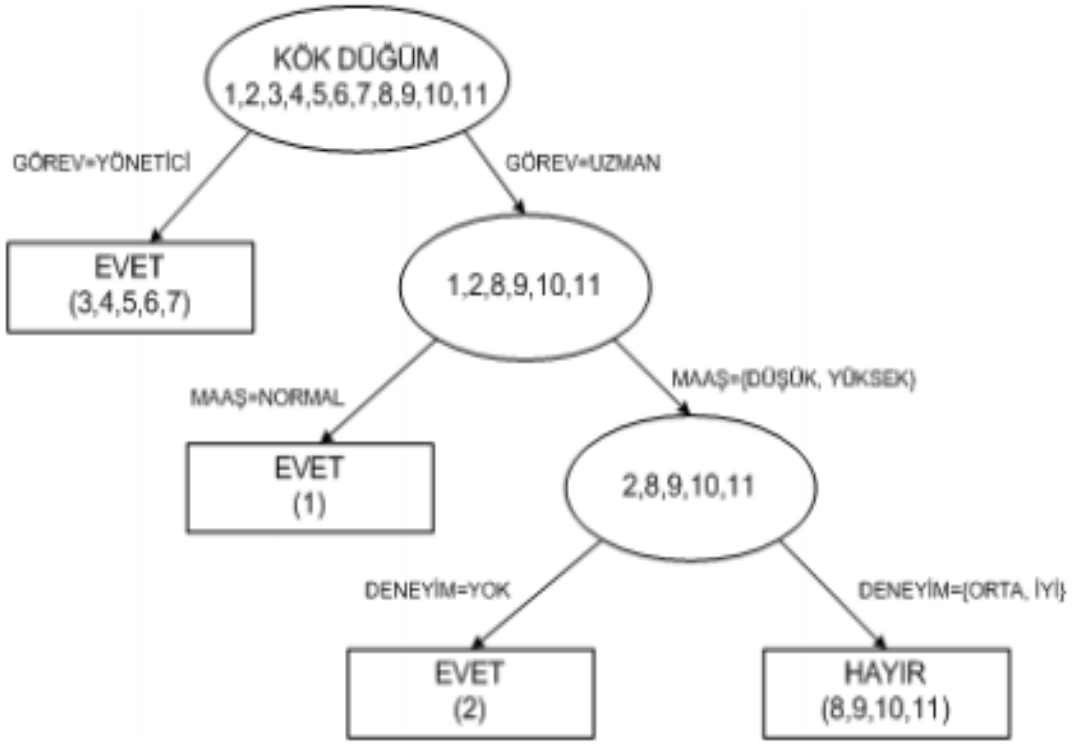
Örnek

- Aynı işlemler ALT DÜĞÜM için tekrarlanır.



Örnek

- Sonuç karar ağacı.



Örnek

- Karar ağacından elde edilen kurallar
  - 1. EĞER (GÖREV = YÖNETİCİ) İSE (MEMNUN = EVET)
  - 2. EĞER (GÖREV = UZMAN) VE (MAAŞ = NORMAL) İSE (MEMNUN =EVET)
  - 3. EĞER (GÖREV = UZMAN) VE (MAAŞ = DÜŞÜK VEYA MAAŞ = YÜKSEK) VE (DENEYİM=YOK) İSE (MEMNUN = EVET)
  - 4. EĞER (GÖREV = UZMAN) VE (MAAŞ = DÜŞÜK VEYA MAAŞ = YÜKSEK) VE (DENEYİM = ORTA VEYA DENEYİM = İYİ) İSE (MEMNUN = HAYIR)

Gini Algoritması

- Gini algoritmasında nitelik değerleri iki parçaya ayrılarak bölümleme yapılır.
- Her bölünme için $Gini_{sol}$ ve $Gini_{sağ}$ değerleri hesaplanır.

$$Gini_{sol} = 1 - \sum_{i=1}^{k} \left(\frac{L_i}{|T_{sol}|}\right)^2 \quad Gini_{sağ} = 1 - \sum_{i=1}^{k} \left(\frac{R_i}{|T_{sağ}|}\right)^2$$

k	Sınıfların Sayısı
T	Bir düğümdeki örnekler
$\|T_{sol}\|$	Sol taraftaki örneklerin sayısı
$\|T_{sağ}\|$	Sağ taraftaki örneklerin sayısı
L_i	Sol taraftaki i kategorisindeki örneklerin sayısı
R_i	Sağ taraftaki i kategorisindeki örneklerin sayısı

- Her j niteliği için n eğitim kümesindeki satır sayısı olmak üzere aşağıdaki bağıntının değeri hesaplanır.

$$Gini_j = \frac{1}{n}(|T_{sol}|Gini_{sol} + |T_{sağ}|Gini_{sağ})$$

Gini Algoritması

- Her j niteliği için n eğitim kümesindeki satır sayısı olmak üzere aşağıdaki bağıntının değeri hesaplanır.

$$Gini_j = \frac{1}{n}(|T_{sol}|Gini_{sol} + |T_{sağ}|Gini_{sağ})$$

- Her j niteliği için hesaplana $Gini_j$ değerleri arasından en küçük olanı seçilir ve bölünme bu öznitelik üzerinden gerçekleştirilir.
- İlk adıma dönülerek işlemler tekrar edilir.

Örnek

SIRA	EĞİTİM	YAŞ	CİNSİYET	SONUÇ
1	ORTA	YAŞLI	ERKEK	EVET
2	İLK	GENÇ	ERKEK	HAYIR
3	YÜKSEK	ORTA	KADIN	HAYIR
4	ORTA	ORTA	ERKEK	EVET
5	İLK	ORTA	ERKEK	EVET
6	YÜKSEK	YAŞLI	KADIN	EVET
7	İLK	GENÇ	KADIN	HAYIR
8	ORTA	ORTA	ERKEK	EVET

	EĞİTİM		YAŞ		CİNSİYET	
SONUÇ	İLK	ORTA,YÜKSEK	GENÇ	ORTA,YAŞLI	KADIN	ERKEK
EVET	1	4	0	5	1	4
HAYIR	2	1	2	1	2	1

BAKIRÇAY ÜNİVERSİTESİ

Örnek

SONUÇ	EĞİTİM		YAŞ		CİNSİYET	
	İLK	ORTA,YÜKSEK	GENÇ	ORTA,YAŞLI	KADIN	ERKEK
EVET	1	4	0	5	1	4
HAYIR	2	1	2	1	2	1

EĞİTİM için

$$Gini_{sol} = 1 - \left[\left(\frac{1}{3}\right)^2 + \left(\frac{2}{3}\right)^2\right] = 0{,}444$$

$$Gini_{sag} = 1 - \left[\left(\frac{4}{5}\right)^2 + \left(\frac{1}{5}\right)^2\right] = 0{,}320$$

Örnek

SONUÇ	EĞİTİM		YAŞ		CİNSİYET	
	İLK	ORTA,YÜKSEK	GENÇ	ORTA,YAŞLI	KADIN	ERKEK
EVET	1	4	0	5	1	4
HAYIR	2	1	2	1	2	1

YAŞ için

$$Gini_{sol} = 1 - \left[\left(\frac{0}{2}\right)^2 + \left(\frac{2}{2}\right)^2\right] = 0$$

$$Gini_{sag} = 1 - \left[\left(\frac{5}{6}\right)^2 + \left(\frac{1}{6}\right)^2\right] = 0{,}278$$

Örnek

SONUÇ	EĞİTİM		YAŞ		CİNSİYET	
	İLK	ORTA,YÜKSEK	GENÇ	ORTA,YAŞLI	KADIN	ERKEK
EVET	1	4	0	5	1	4
HAYIR	2	1	2	1	2	1

CİNSİYET için

$$Gini_{sol} = 1 - \left[\left(\frac{1}{3}\right)^2 + \left(\frac{2}{3}\right)^2\right] = 0{,}444$$

$$Gini_{sag} = 1 - \left[\left(\frac{4}{5}\right)^2 + \left(\frac{1}{5}\right)^2\right] = 0{,}320$$

BAKIRÇAY ÜNİVERSİTESİ

Örnek

Gini değerleri

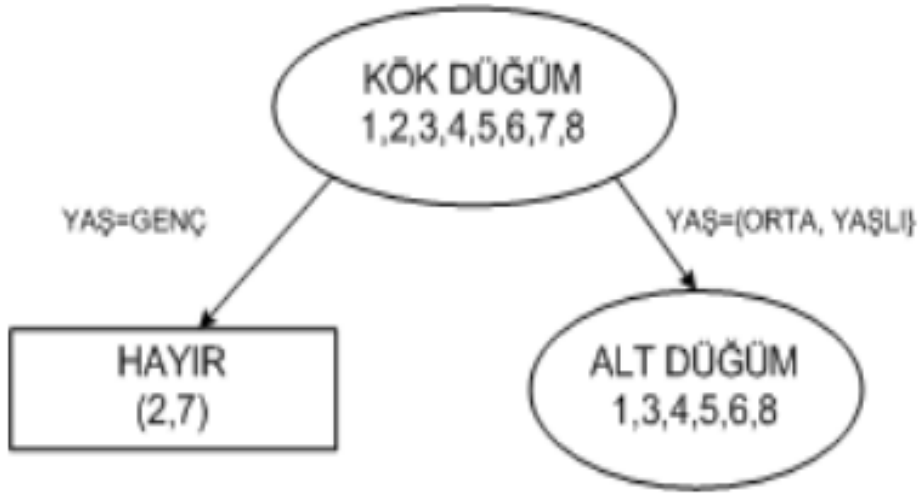
$$Gini_{EGITIM} = \frac{3(0,444) + 5(0,320)}{8} = 0,367$$

$$Gini_{YAS} = \frac{2(0) + 6(0,278)}{8} = 0,209$$

$$Gini_{CINSIYET} = \frac{3(0,444) + 5(0,320)}{8} = 0,367$$

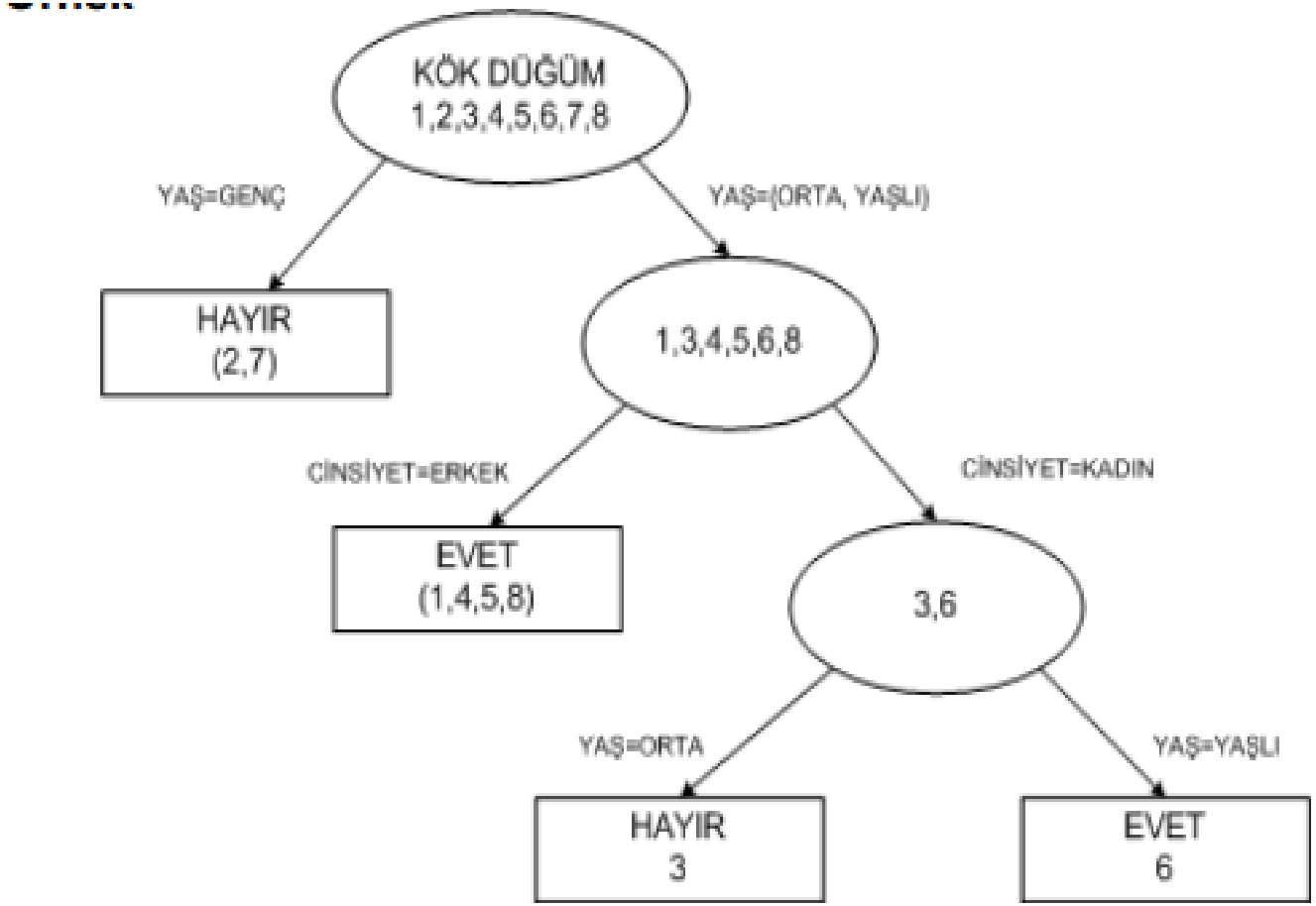
İlk bölünme YAŞ niteliğine göre yapılacaktır.

Örnek



Aynı işlemler ALT DÜĞÜM için tekrarlanır.

Örnek



Örnek

- Karar ağacından elde edilen kurallar
  - 1. EĞER (YAŞ = GENÇ) İSE (SONUÇ = HAYIR)
  - 2. EĞER (YAŞ = ORTA VEYA YAŞ = YAŞLI) VE (CİNSİYET = ERKEK) İSE (SONUÇ = EVET)
  - 3. EĞER (YAŞ = ORTA VEYA YAŞ = YAŞLI) VE (CİNSİYET = KADIN) VE (YAŞ = YAŞLI) İSE (SONUÇ = EVET)
  - 4. EĞER (YAŞ = ORTA VEYA YAŞ = YAŞLI) VE (CİNSİYET = KADIN) VE (YAŞ = ORTA) İSE (SONUÇ = HAYIR)

Sayısal Değerler için Gini Algoritması

MODEL	CİNSİYET	YAŞ	MEMNUN
X5	ERKEK	21	HAYIR
X3	KADIN	19	EVET
X5	ERKEK	22	HAYIR
X3	ERKEK	21	EVET
X3	ERKEK	30	EVET
X3	KADIN	60	HAYIR
X3	KADIN	45	HAYIR
X3	ERKEK	55	HAYIR

C4.5 algoritmasına benzer bir yol izlenir. Her bir sayısal değer eşik değeri olarak belirlenerek gini ölçütü hesaplanır. İşlemler sonucunda en küçük gini ölçütünü sağlayacak eşik değeri seçilir.

Sayısal Değerler için Gini Algoritması

MEMNUN	MODEL		CİNSİYET	
	X5	X3	ERKEK	KADIN
EVET	0	3	2	1
HAYIR	2	3	3	2

$$Gini_{X5} = 1 - \left[\left(\frac{0}{2}\right)^2 + \left(\frac{2}{2}\right)^2\right] = 0$$

$$Gini_{X3} = 1 - \left[\left(\frac{3}{6}\right)^2 + \left(\frac{3}{6}\right)^2\right] = 0.5$$

$$Gini_{MODEL} = \frac{|T_{X5}|Gini_{X5} + |T_{X3}|Gini_{X3}}{n} = \frac{(2)(0) + (6)(0.5)}{8} = 0.375$$

BAKIRÇAY ÜNİVERSİTESİ

Sayısal Değerler için Gini Algoritması

MEMNUN	MODEL		CİNSİYET	
	X5	X3	ERKEK	KADIN
EVET	0	3	2	1
HAYIR	2	3	3	2

$$Gini_{ERKEK} = 1 - \left[\left(\frac{2}{5}\right)^2 + \left(\frac{3}{5}\right)^2\right] = 0.48$$

$$Gini_{KADIN} = 1 - \left[\left(\frac{1}{3}\right)^2 + \left(\frac{2}{3}\right)^2\right] = 0.45$$

$$Gini_{CİNSİYET} = \frac{|T_{ERKEK}|Gini_{ERKEK} + |T_{KADIN}|Gini_{KADIN}}{n} = \frac{(5)(0.48) + (3)(0.45)}{8} = 0.469$$

BAKIRÇAY ÜNİVERSİTESİ

Sayısal Değerler için Gini Algoritması

YAŞ	MEMNUN
21	HAYIR
19	EVET
22	HAYIR
21	EVET
30	EVET
60	HAYIR
45	HAYIR
55	HAYIR

MEMNUN	YAŞ	
	≤19	>19
EVET	1	2
HAYIR	0	5

$$Gini_{\leq 19} = 1 - \left[\left(\frac{1}{1}\right)^2 + \left(\frac{0}{1}\right)^2\right] = 0$$

$$Gini_{> 19} = 1 - \left[\left(\frac{2}{7}\right)^2 + \left(\frac{5}{7}\right)^2\right] = 0.408$$

$$Gini_{YAŞ = 19} = \frac{(1)(0) + (7)(0.408)}{8} = 0.357$$

BAKIRÇAY ÜNİVERSİTESİ

Sayısal Değerler için Gini Algoritması

YAŞ	MEMNUN
21	HAYIR
19	EVET
22	HAYIR
21	EVET
30	EVET
60	HAYIR
45	HAYIR
55	HAYIR

MEMNUN	YAŞ	
	≤21	>21
EVET	2	1
HAYIR	1	4

$$Gini_{\leq 21} = 1 - \left[\left(\frac{2}{3}\right)^2 + \left(\frac{1}{3}\right)^2\right] = 0.445$$

$$Gini_{> 21} = 1 - \left[\left(\frac{1}{5}\right)^2 + \left(\frac{4}{5}\right)^2\right] = 0.32$$

$$Gini_{YAŞ = 21} = \frac{(3)(0.445) + (5)(0.32)}{8} = 0.369$$

BAKIRÇAY ÜNİVERSİTESİ

Sayısal Değerler için Gini Algoritması

YAŞ	MEMNUN
21	HAYIR
19	EVET
22	HAYIR
21	EVET
30	EVET
60	HAYIR
45	HAYIR
55	HAYIR

MEMNUN	YAŞ	
	≤22	>22
EVET	2	1
HAYIR	2	3

$$Gini_{\leq 22} = 1 - \left[\left(\frac{2}{4}\right)^2 + \left(\frac{2}{4}\right)^2\right] = 0.5$$

$$Gini_{> 22} = 1 - \left[\left(\frac{1}{4}\right)^2 + \left(\frac{3}{4}\right)^2\right] = 0.375$$

$$Gini_{YAŞ = 22} = \frac{(4)(0.5) + (4)(0.375)}{8} = 0.438$$

BAKIRÇAY ÜNİVERSİTESİ

Sayısal Değerler için Gini Algoritması

YAŞ	MEMNUN
21	HAYIR
19	EVET
22	HAYIR
21	EVET
30	EVET
60	HAYIR
45	HAYIR
55	HAYIR

MEMNUN	YAŞ	
	≤30	>30
EVET	3	0
HAYIR	2	3

$$Gini_{\leq 30} = 1 - \left[\left(\frac{3}{5}\right)^2 + \left(\frac{2}{5}\right)^2\right] = 0.48$$

$$Gini_{> 30} = 1 - \left[\left(\frac{0}{3}\right)^2 + \left(\frac{3}{3}\right)^2\right] = 0$$

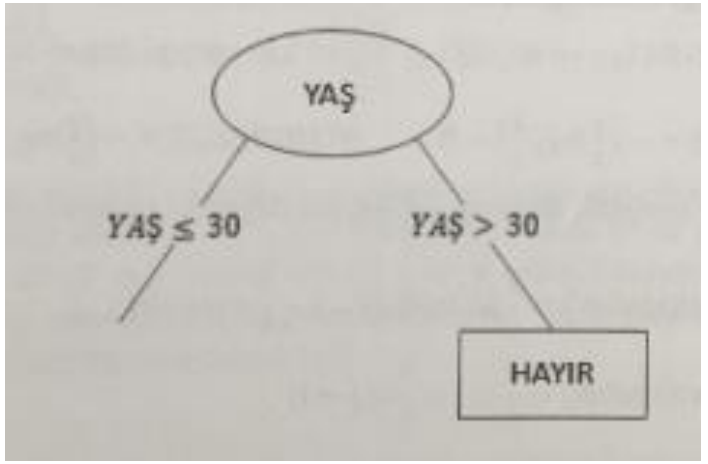
$$Gini_{YAŞ = 30} = \frac{(5)(0.48) + (3)(0)}{8} = 0.3$$

BAKIRÇAY ÜNİVERSİTESİ

Sayısal Değerler için Gini Algoritması

YAŞ	Gini Ölçütü
19	0,357
21	0,369
22	0,438
30	**0,3**
45	0,375
55	0,429
60	0,469

Öznitelik	Gini Ölçütü
MODEL	0,375
CİNSİYET	0,469
YAŞ	0.3



Regresyon Ağaçları

Gözlemlerin iki dala ayrılması ve her bir dal için “kareli hata” değerinin hesaplanması esasına dayanmaktadır.
Her bir t düğümü için n(t) söz konusu düğümün eleman sayısını göstermek üzere

$$R(t) = \frac{1}{n(t)} \sum_{X_i \in t} \left(y_i - \bar{y}(t)\right)^2$$

Ağaç üzerinde ikili bölünme yapıldığı varsayılırsa, t_{sol} ve $t_{sağ}$ biçiminde iki dala ayrılacaktır. Her bir bölünme için:

$$R(t_{sol}) = \frac{1}{n(t_{sol})} \sum_{X_i \in t} \left(y_i - \bar{y}(t_{sol})\right)^2$$

$$R(t_{sağ}) = \frac{1}{n(t_{sağ})} \sum_{X_i \in t} \left(y_i - \bar{y}(t_{sağ})\right)^2$$

Hesaplanacak ve minimum($R(t_{sol})+R(t_{sağ})$) değeri bölmenin yapılacağı noktayı belirleyecektir.

Örnek

Örnek No	X1	X2	Y
1	12	34	44
2	18	25	35
3	11	18	30
4	9	23	45
5	15	31	43

X1 niteliği 1. eleman göz önüne alınarak. X1 ≤ 12 için ikili bölme:

t_{sol}	$t_{sağ}$
44	35
30	43
45	

Örnek

t_{sol}	$t_{sağ}$
44	35
30	43
45	

$$\bar{y}(t_{sol}) = \frac{44 + 30 + 45}{3} = 39.67$$

$$\bar{y}(t_{sağ}) = \frac{35 + 43}{2} = 39$$

$$R(t_{sol}) = \frac{(44 - 39.67)^2 + (30 - 39.67)^2 + (45 - 39.67)^2}{3} = 46{,}89$$

$$R(t_{sağ}) = \frac{(35 - 39)^2 + (43 - 39)^2}{2} = 16$$

$R(t_{sol}) + R(t_{sağ}) = 46{,}89 + 16 = 62{,}89$

Örnek

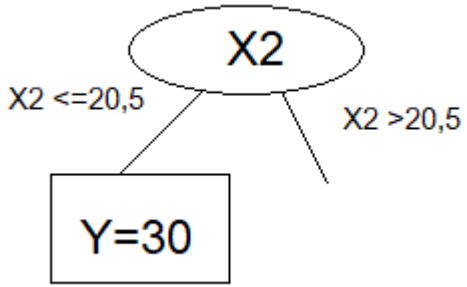
X1 ≤ 12 için yapılan hesaplama bütün ihtimaller için gerçekleştirilir.

Koşul	$R(t_{sol}) + R(t_{sağ})$
X1 ≤ 12	62,89
X1 ≤ 18	34,64
X1 ≤ 11	72,47
X1 ≤ 9	33,50
X1 ≤ 15	37,25
X2 ≤ 34	34,64
X2 ≤ 25	39,14
X2 ≤ 18	15,69
X2 ≤ 23	72,47
X2 ≤ 31	36,69

Bu tabloda en düşük $R(t_{sol}) + R(t_{sağ})$ değerini sağlayan noktanın X2 ≤ 18 olduğu görülmektedir. Bölünme noktası olarak X2=18 ve bir altındaki değer olan X2=23 noktalarının ortası alınacaktır.

Bölünme Noktası = (18 + 23) / 2 = 20.5

Örnek



Elde edilen kesme noktası göz önüne alınarak gözlem tablosu yeniden düzenlenecek ve aynı işlemler tekrar edilecektir.

Örnek No	X1	X2	Y
1	12	34	44
2	18	25	35
4	9	23	45
5	15	31	43

Bir Modelin Seçilmesi için Gereken Kriterler

Kesinlik (Accuracy) : Bir veri kümesi modellenmesinde yeni veri girişi yapıldığında sistemin tutarlı hareket etmesi durumudur.

Hız (Speed) : Veri kümesinin değerlendirilmesinin , doğru sonuçlar ortaya çıkarmasının hızı da önemli bir kriterdir. Yeri geldiğinde bir modelin seçilmesi için başarı durumuna bakılırken yeri geldiğinde hız daha büyük bir etkendir modelin seçilmesinde .

Sağlamlık (Robustness) : Bir modelin eksik ve gürültülü verilerden ne kadar etkilendiği, bu verilere karşı olan hassaslığı o modelin seçilmesi konusunda önemlidir.

Ölçeklenebilirlik (Scalability) : Eldeki veriye göre sistemin başarısının artıp azalmamasıdır. Veri girişi sağlandığında başarı oranının düşmemesi gerekmektedir.

Bir Modelin Seçilmesi için Gereken Kriterler

Yorumlanabilirlik (Interpretability) : Bir algoritmanın anlaşılıp anlatılabilir olması önemlidir. Bir algoritmanın kolay anlaşılabilir olması tercih edilmesi konusunda önemli bir kriterdir. Anlaşılan bir algoritmanın sonuçları da analiz edilmesi kolaylaşacaktır.

Akan bir veri trafiğine sahip (twitter gibi) olunduğu düşünüldüğünde ve bu verinin anlık bir şekilde sınıflandırılması ve model oluşturulması gerektiği durumlarda önem sırasında hız önemli bir etken iken bir hastalık durumunun analiz edildiği bir durum karşısında başarı kriterinin önemi artmaktadır. Görüldüğü üzere çalışılan ortamlara göre modelin seçilme kriteri değişebilmektedir.

Regresyon başarı metrikleri

https://scikit-learn.org/stable/modules/classes.html#regression-metrics

metrics.explained_variance_score(y_true, ...)	Explained variance regression score function.
metrics.max_error(y_true, y_pred)	The max_error metric calculates the maximum residual error.
metrics.mean_absolute_error(y_true, y_pred, *)	Mean absolute error regression loss.
metrics.mean_squared_error(y_true, y_pred, *)	Mean squared error regression loss.
metrics.mean_squared_log_error(y_true, y_pred, *)	Mean squared logarithmic error regression loss.
metrics.median_absolute_error(y_true, y_pred, *)	Median absolute error regression loss.
metrics.mean_absolute_percentage_error(...)	Mean absolute percentage error (MAPE) regression loss.
metrics.r2_score(y_true, y_pred, *[, ...])	R^2 (coefficient of determination) regression score function.
metrics.mean_poisson_deviance(y_true, y_pred, *)	Mean Poisson deviance regression loss.
metrics.mean_gamma_deviance(y_true, y_pred, *)	Mean Gamma deviance regression loss.
metrics.mean_tweedie_deviance(y_true, y_pred, *)	Mean Tweedie deviance regression loss.
metrics.d2_tweedie_score(y_true, y_pred, *)	D^2 regression score function, fraction of Tweedie deviance explained.
metrics.mean_pinball_loss(y_true, y_pred, *)	Pinball loss for quantile regression.
metrics.d2_pinball_score(y_true, y_pred, *)	D^2 regression score function, fraction of pinball loss explained.
metrics.d2_absolute_error_score(y_true, ...)	D^2 regression score function, fraction of absolute error explained.

BAKIRÇAY ÜNİVERSİTESİ

Regresyon başarı metrikleri

```
from sklearn.metrics import mean_squared_error
y_true = [3, -0.5, 2, 7]
y_pred = [2.5, 0.0, 2, 8]
mse = mean_squared_error(y_true, y_pred,squared=True)
print(mse)
```

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

```
from sklearn.metrics import mean_squared_error
y_true = [3, -0.5, 2, 7]
y_pred = [2.5, 0.0, 2, 8]
rmse = mean_squared_error(y_true, y_pred,squared=False)
print(rmse)
```

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Regresyon başarı metrikleri

R^2, toplam açıklanan varyansın (SS_model) toplam varyansa (SS_total) oranı olarak hesaplanır.

$R^2 = 1$: Model veriyi mükemmel bir şekilde açıklıyor, yani bağımlı değişkenin varyansı tamamen bağımsız değişkenlerle açıklanabiliyor.
$R^2 = 0$: Model, bağımlı değişkenin varyansını hiç açıklayamıyor, yani tahminler tamamen ortalamaya dayalı.
$0 < R^2 < 1$: Model bağımlı değişkenin varyansının bir kısmını açıklıyor ancak mükemmel değil.

```
from sklearn.metrics import r2_score
y_true = [3, -0.5, 2, 7]
y_pred = [2.5, 0.0, 2, 8]
r2 = r2_score(y_true, y_pred)
print(r2)
```

$$R^2 = 1 - \frac{\mathrm{SS}_{\mathrm{residual}}}{\mathrm{SS}_{\mathrm{total}}}$$

$$\mathrm{SS}_{\mathrm{total}} = \sum_{i=1}^{n} (y_i - \overline{y})^2$$

$$\mathrm{SS}_{\mathrm{residual}} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Burada:

- y_i: Gerçek değer
- $\hat{y}_i$: Modelin tahmin ettiği değer
- $\overline{y}$: Gerçek değerlerin ortalaması

Sınıflandırma başarı metrikleri

https://scikit-learn.org/stable/modules/model_evaluation.html#classification-metrics

Scoring	Function	Comment
Classification		
'accuracy'	`metrics.accuracy_score`	
'balanced_accuracy'	`metrics.balanced_accuracy_score`	
'top_k_accuracy'	`metrics.top_k_accuracy_score`	
'average_precision'	`metrics.average_precision_score`	
'neg_brier_score'	`metrics.brier_score_loss`	
'f1'	`metrics.f1_score`	for binary targets
'f1_micro'	`metrics.f1_score`	micro-averaged
'f1_macro'	`metrics.f1_score`	macro-averaged

Karmaşıklık Matrisi (Confusion Matrix)

Başvuru	EĞİTİM	YAŞ	CİNSİYET	KABUL	TAHMİN
1	ORTA	YAŞLI	ERKEK	EVET	EVET
2	İLK	GENÇ	ERKEK	HAYIR	HAYIR
3	YÜKSEK	ORTA	KADIN	HAYIR	EVET
4	ORTA	ORTA	ERKEK	EVET	EVET
5	İLK	ORTA	ERKEK	EVET	EVET
6	YÜKSEK	YAŞLI	KADIN	EVET	EVET
7	İLK	GENÇ	KADIN	HAYIR	HAYIR
8	ORTA	ORTA	KADIN	EVET	EVET

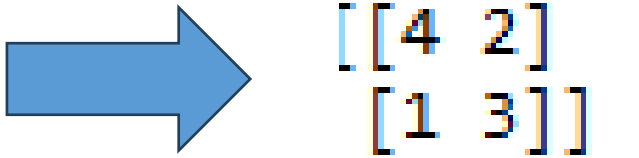
		TAHMİN		
		EVET	HAYIR	TOPLAM
GERÇEK DEĞER	EVET	TP=5	FN=0	5
	HAYIR	FP=1	TN=2	3
	TOPLAM	6	2	8

TP => True Positive
TN => True Negative
FP => False Positive
FN => False Negative

BAKIRÇAY ÜNİVERSİTESİ

Karmaşıklık Matrisi (Confusion Matrix)

```
from sklearn.metrics import confusion_matrix
y_true = [1, 0, 1, 0, 1, 1, 0, 0, 0, 0]
y_pred = [0, 1, 1, 0, 1, 1, 1, 0, 0, 0]
cm = confusion_matrix(y_true, y_pred)
print(cm)
```



```
[[4 2]
 [1 3]]
```

		TAHMİN		
		0	1	TOPLAM
GERÇEK DEĞER	0	TP=4	FN=2	6
	1	FP=1	TN=3	4
	TOPLAM	5	5	10

BAKIRÇAY ÜNİVERSİTESİ

Sınıflandırma Başarı Ölçütleri

		TAHMİN		
		0	1	TOPLAM
GERÇEK DEĞER	0	TP=4	FN=2	6
	1	FP=1	TN=3	4
	TOPLAM	5	5	10

$$Doğruluk\ (Accuracy(ACC)) = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Hata\ Oranı\ (Error\ Rate(ER)) = 1\ - Doğruluk = \frac{FN + FP}{TP + TN + FP + FN}$$

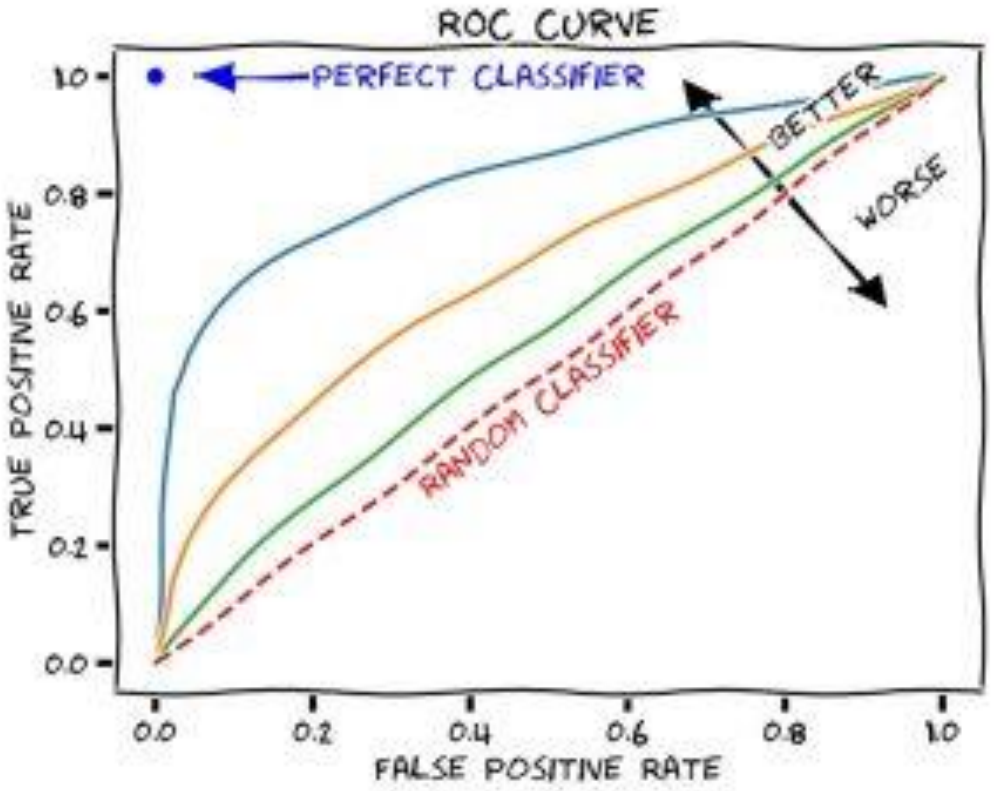
$$Duyarlılık\ (sensitivity, recall, hit\ rate, true\ positive\ rate\ (TPR)) = \frac{TP}{TP + FN}$$

$$Özgünlük(specificity, selectivity\ or\ true\ negative\ rate\ (TNR)) = \frac{TN}{FP + TN}$$

$$Hassaslık\ (precision\ or\ positive\ predictive\ value\ (PPV)) = \frac{TP}{TP + FP}$$

$$\text{F–Ölçütü (f1–score)} = 2\ x\ \frac{PPV\ x\ TPR}{PPV + TPR}$$

Sınıflandırma Başarı Ölçütleri



$$\text{True Positive Rate} = \text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{False Positive Rate} = (1 - \text{Specificity}) = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}}$$

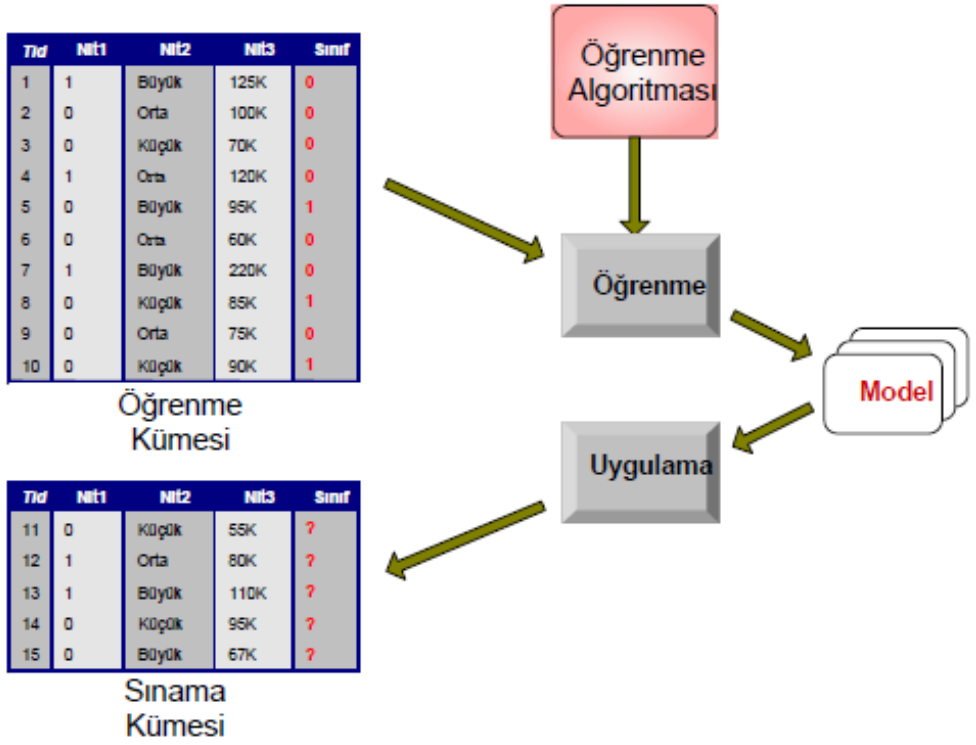
Lojistik regresyon gibi modellerde farklı eşik değerleri için TPR ve FPR değerleri hesaplanarak grafik üzerinde görselleştirilir.

Eğri altında kalan alan (AUC) ne kadar büyük ise o kadar başarılı bir sınıflandırma yapılmıştır.

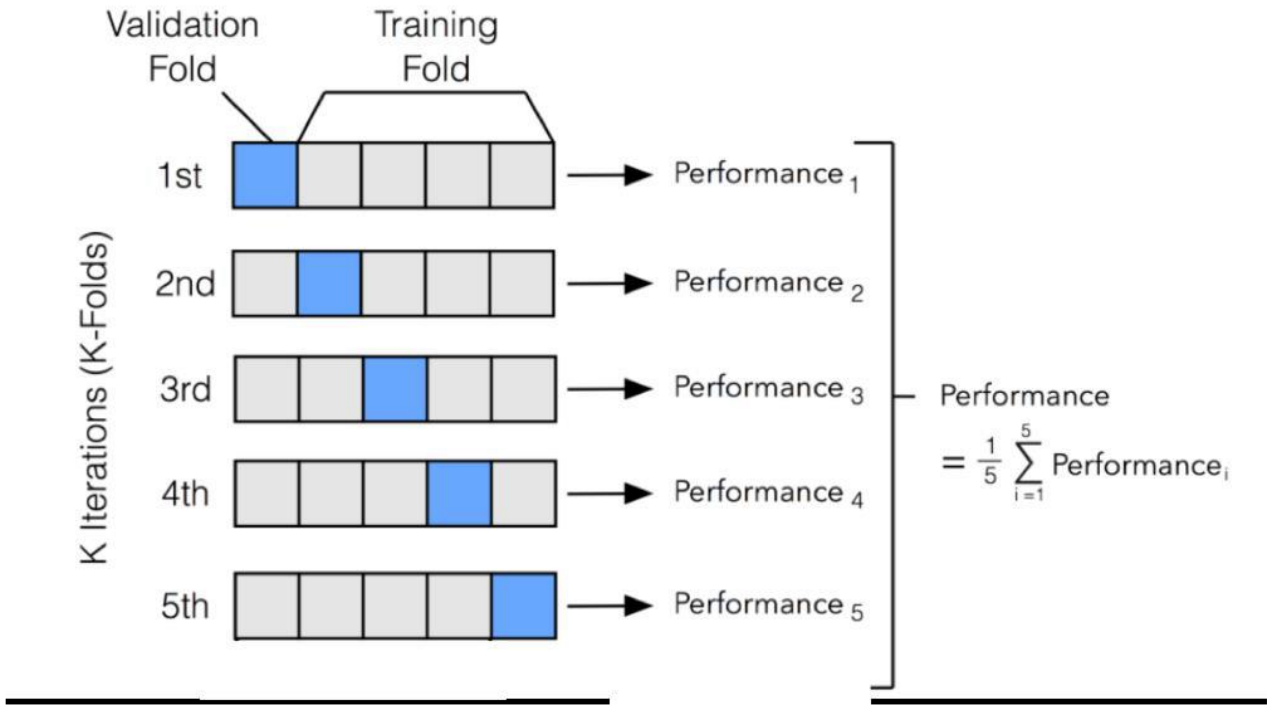
```
import numpy as np
from sklearn import metrics
y = np.array([1, 1, 0, 1])
scores = np.array([0.1, 0.4, 0.35, 0.8])
fpr, tpr, thresholds = metrics.roc_curve(y, scores)
```

Model Başarılarını ölçmek için kullanılan teknikler

- Eğitim ve Test verisi olarak bölme



- Çapraz Doğrulama



Model Başarılarını ölçmek için kullanılan teknikler

- Çapraz Doğrulama :

K-Fold Cross Validation, sınıflandırma modellerinin değerlendirilmesi ve modelin eğitilmesi için veri setini parçalara ayırma yöntemlerinden biridir.

Elimizde bin kayıtlık bir veri seti olsun. Biz bu veri setinin bir kısmı ile modelimizi eğitmek, bir kısmı ile eğittiğimiz modelimizin başarısını değerlendirmek istiyoruz. Basit yaklaşım; %70'ini eğitim için, %30'unu de test için ayırmaktır. Ancak burada veri parçalanırken verinin dağılımına bağlı olarak modelin eğitim ve testinde bazı sapmalar (bias) ve hatalar oluşabilir. k-fold cross validation, veriyi belirlenen bir k sayısına göre eşit parçalara böler, her bir parçanın hem eğitim hem de test için kullanılmasını sağlar, böylelikle dağılım ve parçalanmadan kaynaklanan sapma ve hataları asgariye indirir. Ancak modeli k kadar eğitmek ve test etmek gibi ilave bir veri işleme yük ve zamanı ister. Bu durum eğitim ve testi kısa süren küçük ve orta hacimli veriler için sorun olmasa da büyük hacimli veri setlerinde hesaplama ve zaman yönünden maliyetli olabili

Model Başarılarını ölçmek için kullanılan teknikler

- Çapraz Doğrulama :