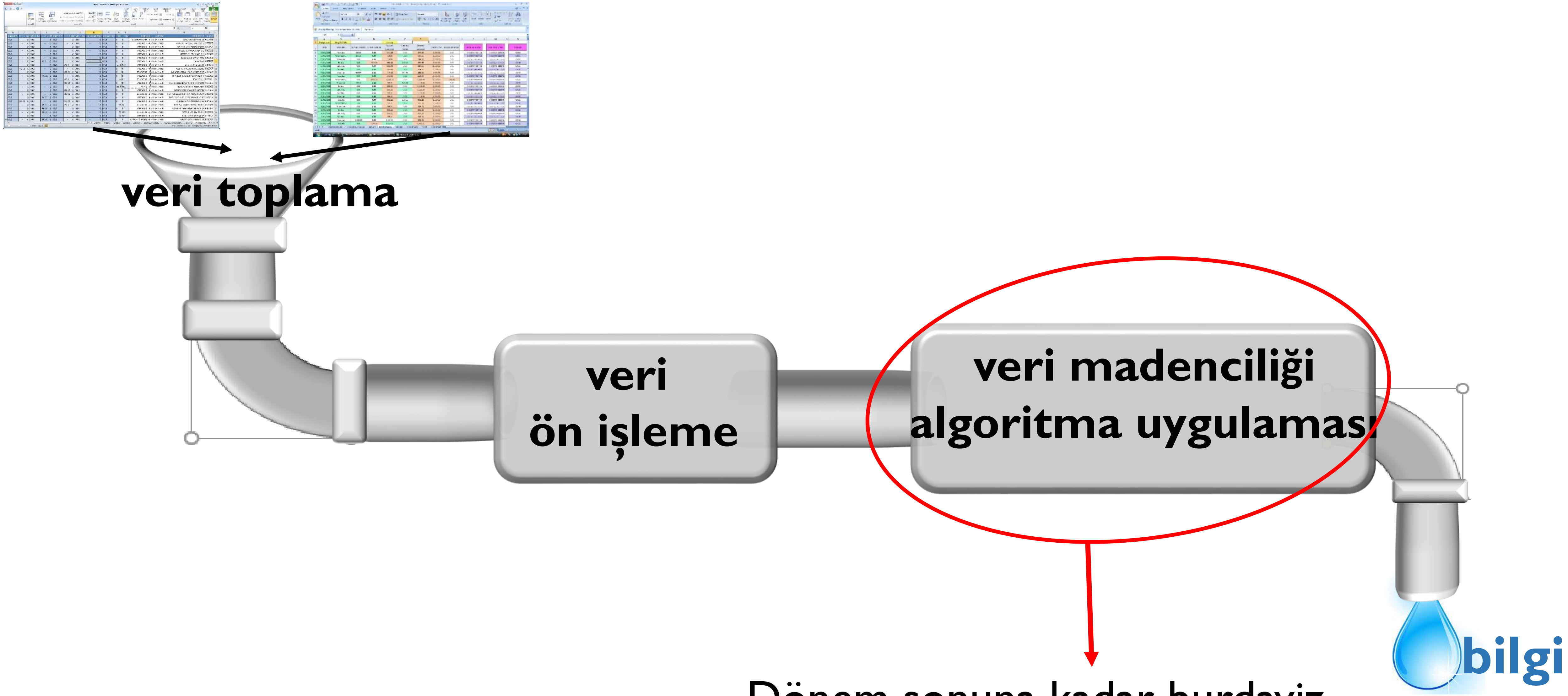
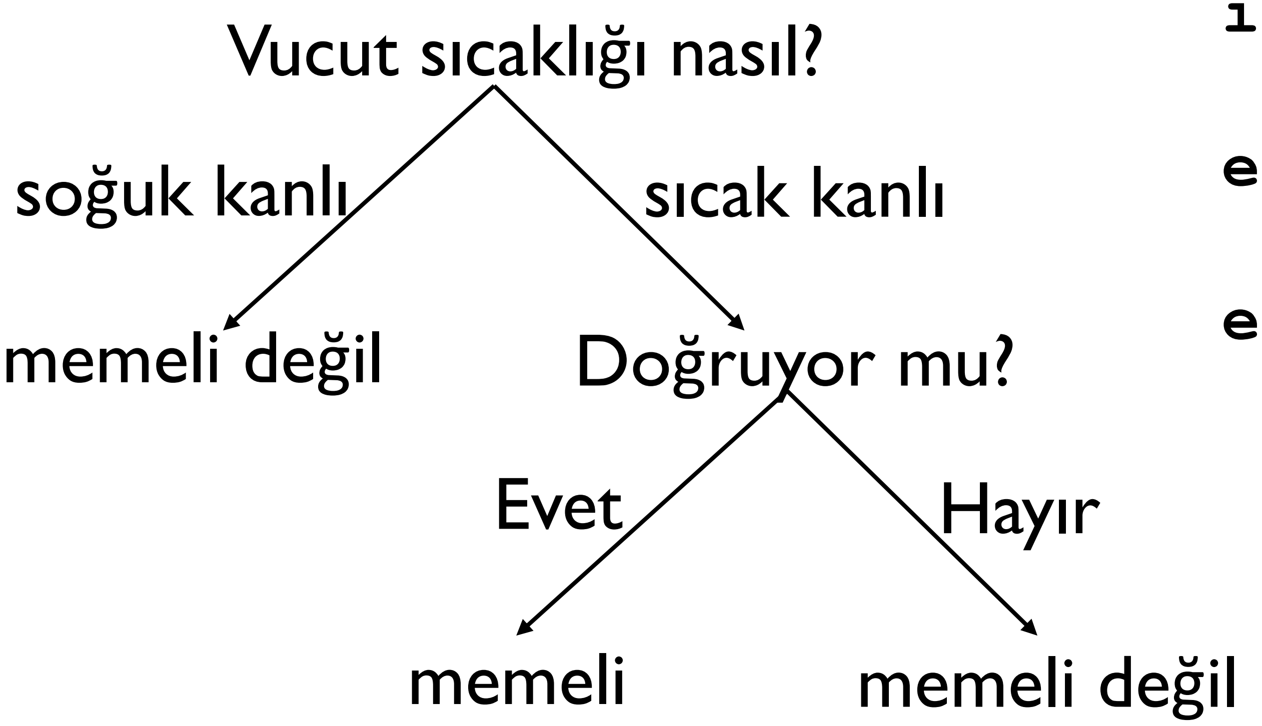
VERİ MADENCİLİĞİ

Fırat İsmailoğlu, PhD

Karar Ağaçları



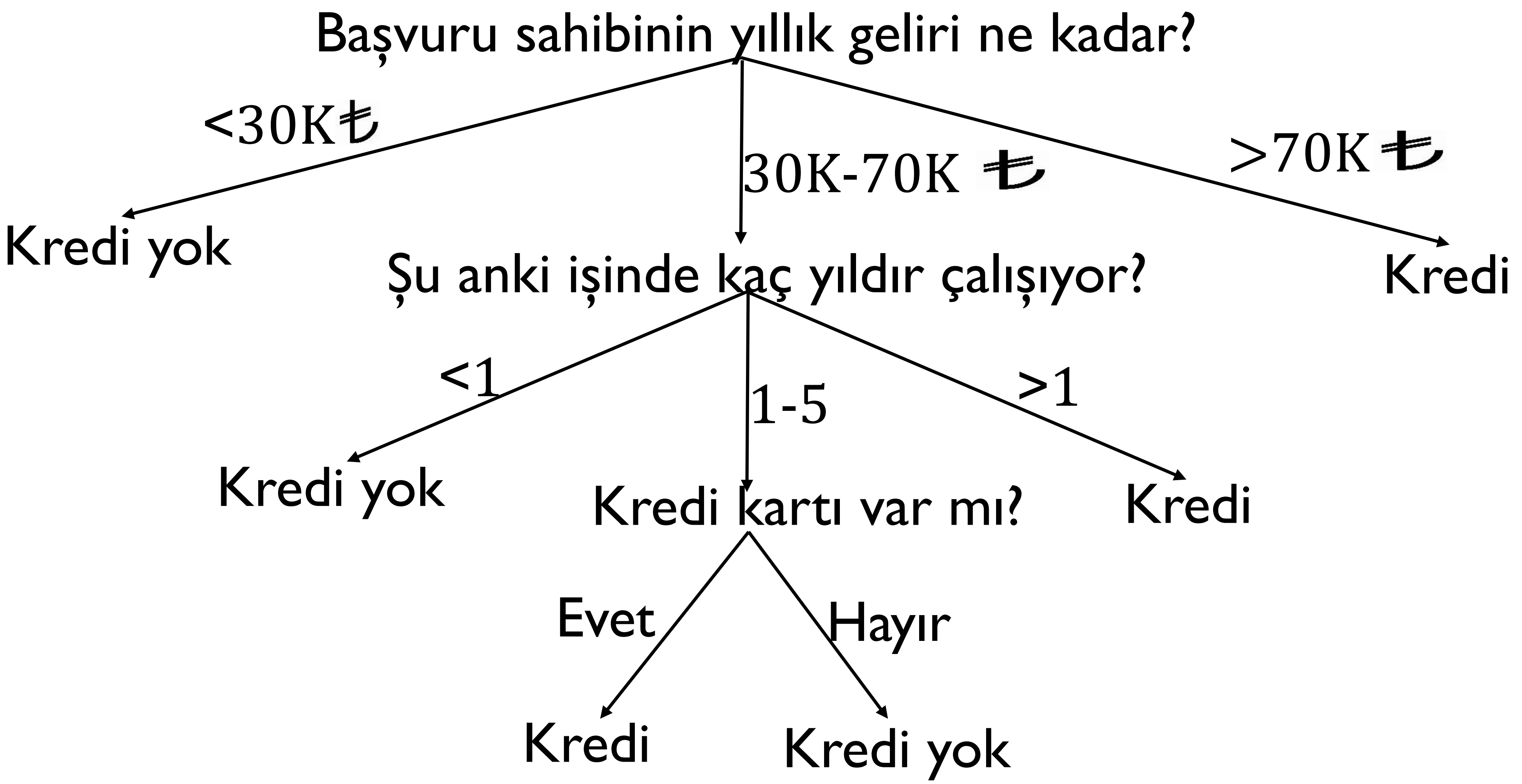
Dönem sonuna kadar burdayiz.




```
if vucutSicakligi==soguk{
    return memeliDegil;
else if doguruyorMu==evet{
    return memeli;
else
    return memeliDegil;}}
```

En önemliden en önemsize doğru ard arda sorular sorarak bir omurgalının memeli olup olmadığına karar veriyoruz.

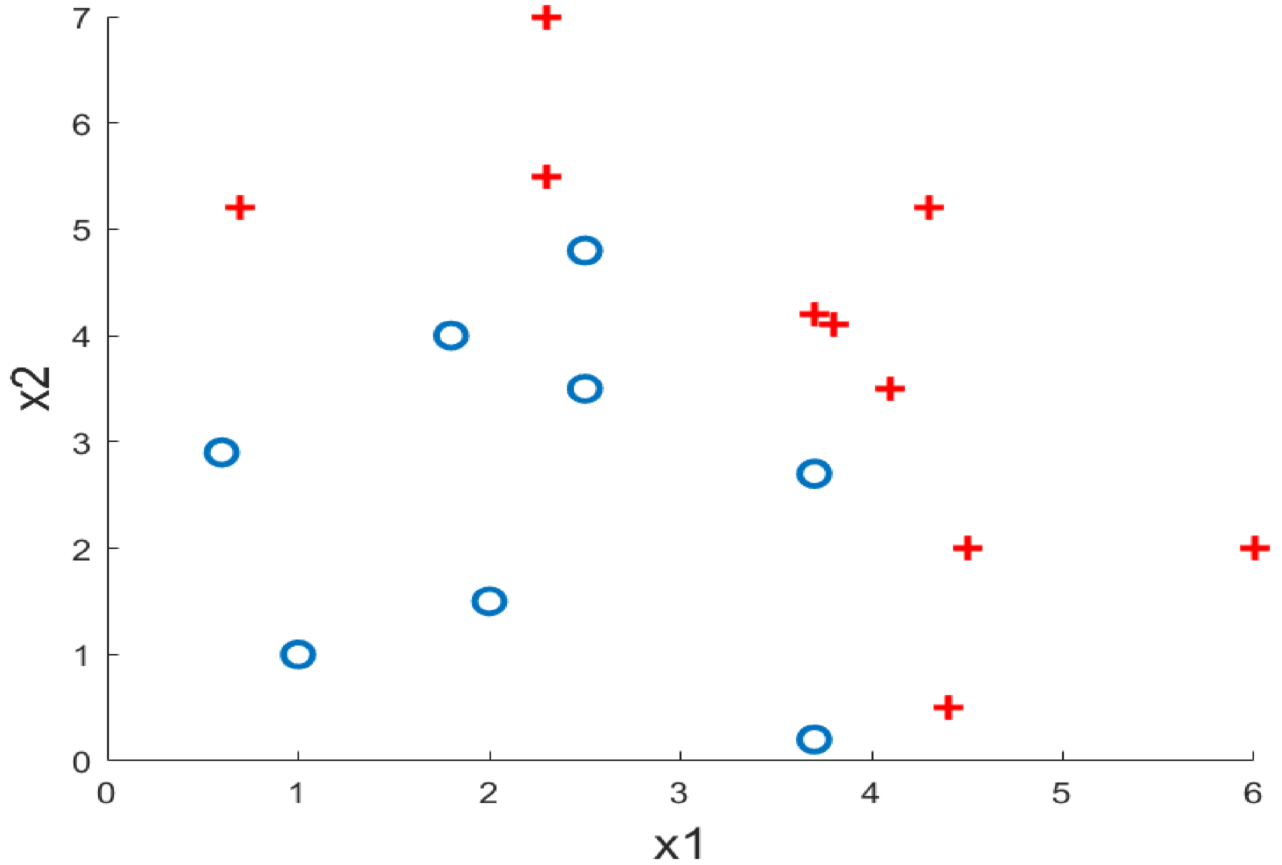
Bir başka deyişle test örnegi en yukaridan aşagiya dogru yuvarlanir. Örnek bir sınıfa (memeli, memeli degil) geldiginde durur. Bu sinif onun tahmin edilen sinifi olur.

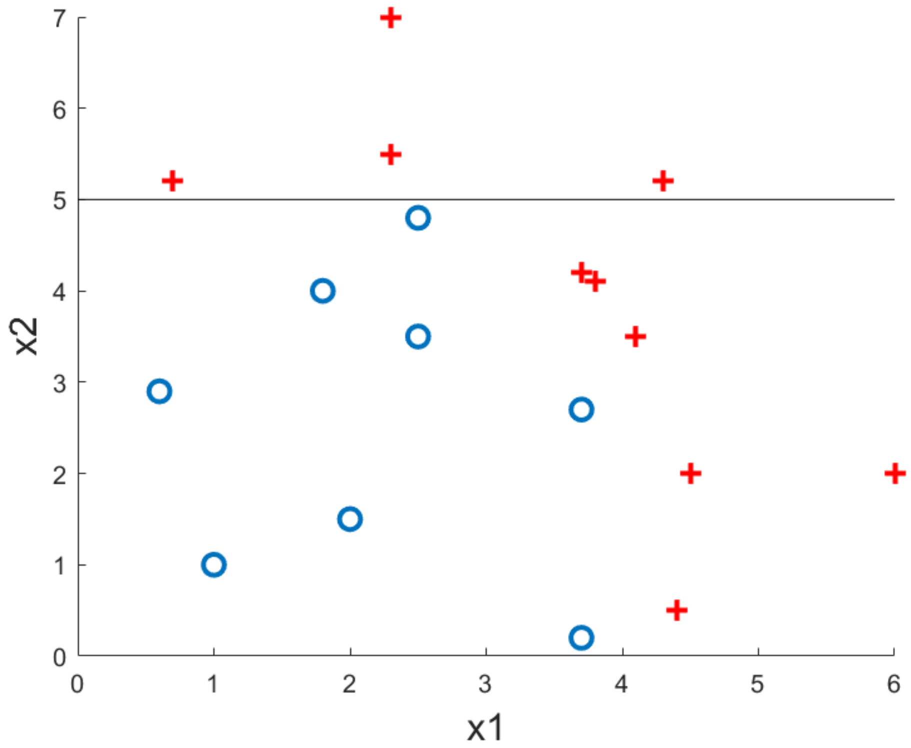


Kredi verilip verilmeyecegine karar veren mekanizma.

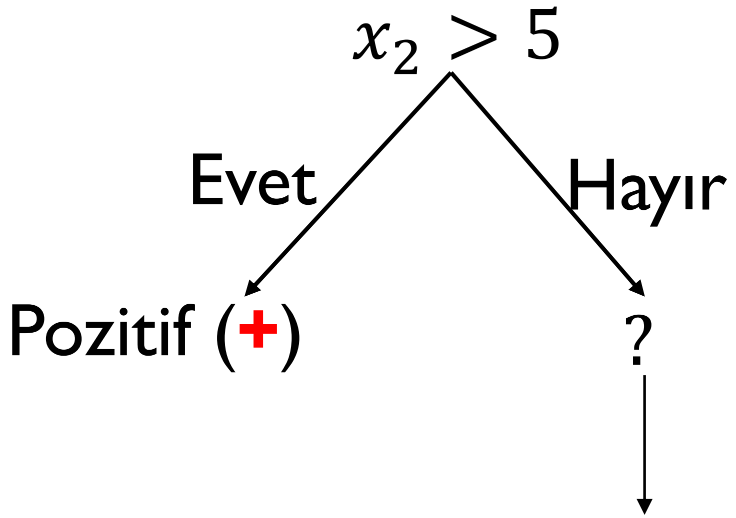
Karar ağaçları giris uzayinda dogrusal olarak ayrilmayan (yani bir hiperdüzlemle ayrilmayan) sınıfları birbirinden ayırmada kullanilan yöntemlerden biridir.

Örnek olarak bir giriş uzayında pozitif (+) ve negatif (o) örnekler asagidaki gibi dagilmis olsun. Burada pozitif ve negatif örnekleri bir hiperduzlemle birbirinden ayırmak mumkun değildir.

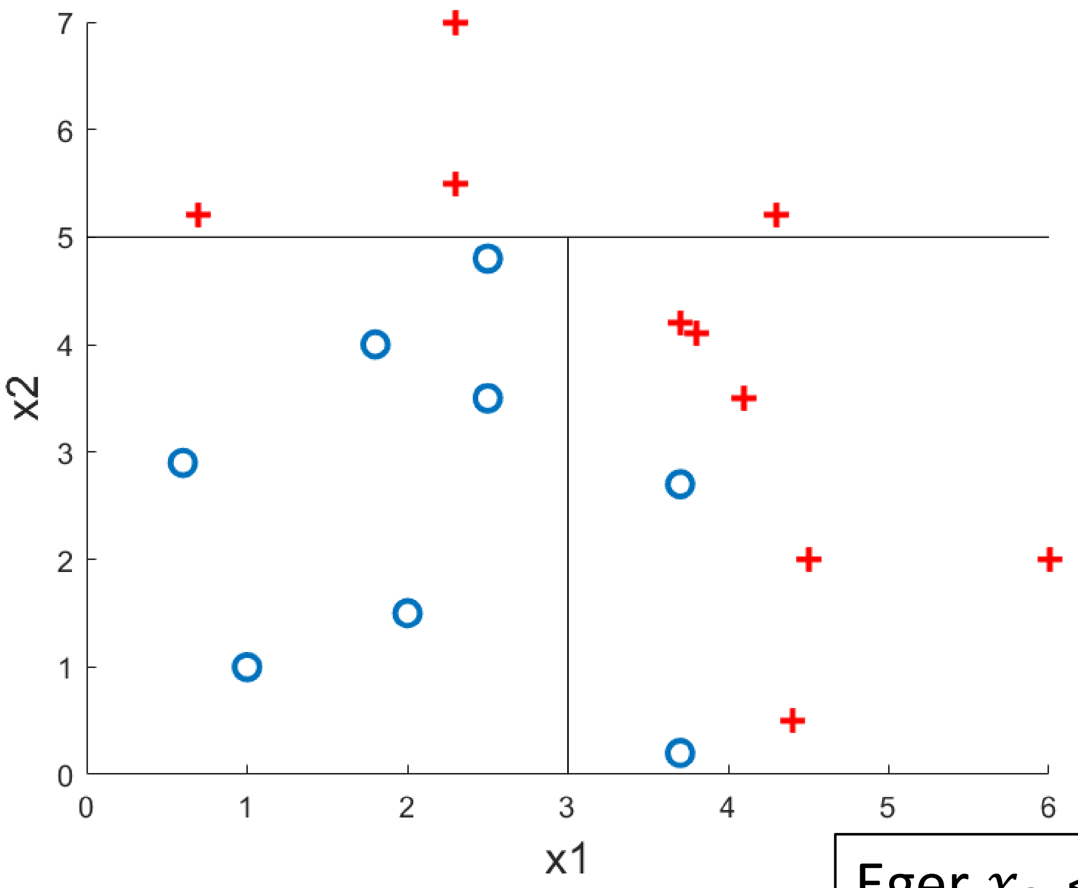


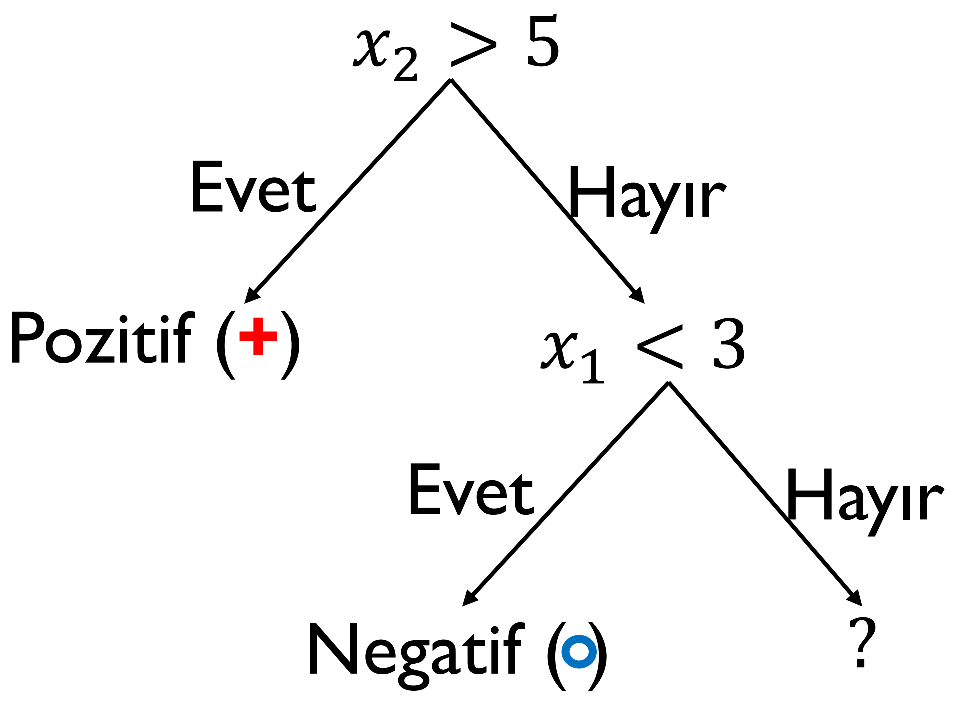


$x_2 = 5$ doğrusu ile uzayi iki parçaya boluyoruz. Oyleki yukarda kalan kismin tamami pozitif orneklerden oluşuyor. Bu durumda bir ornek için eger $x_2 > 5$ ise direkt bu ornek pozitiftir diyebiliriz.

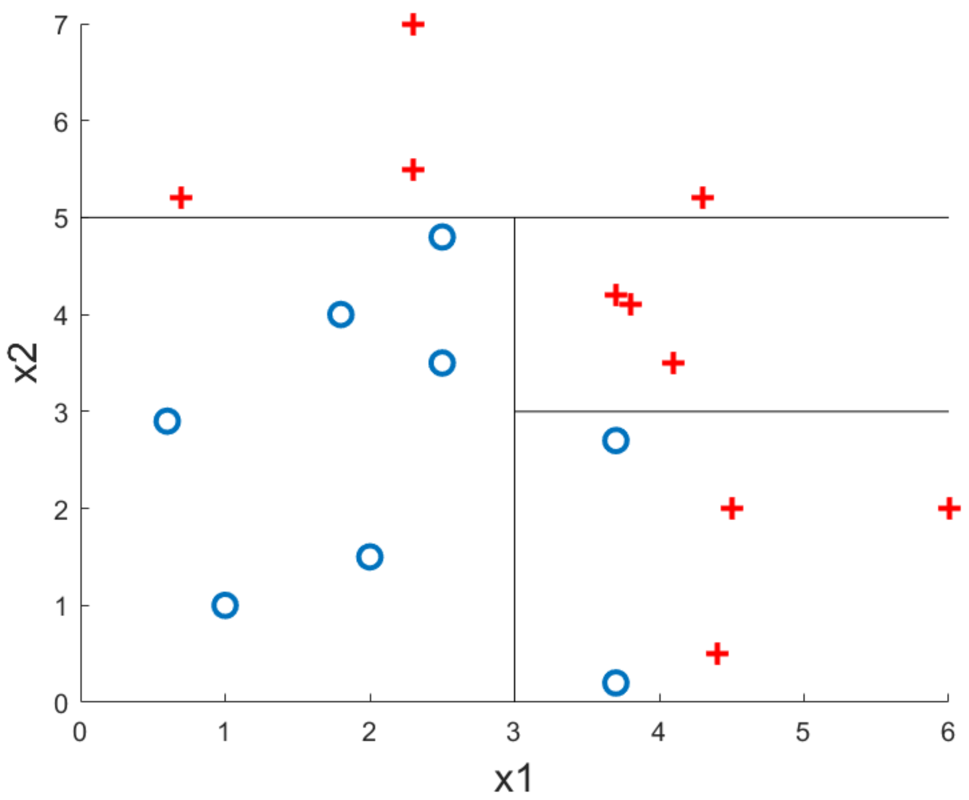


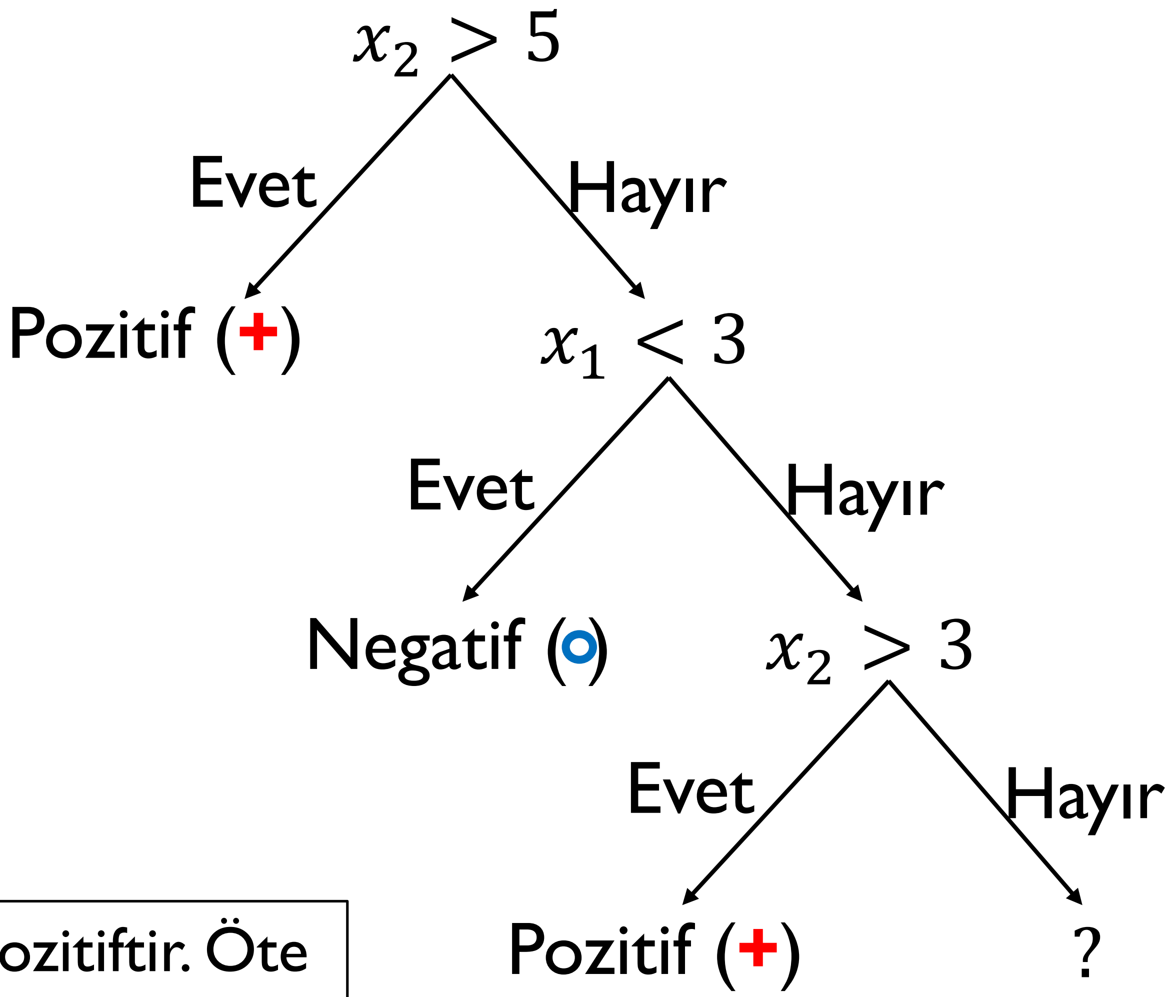
$x_2 < 5$ olan orneklerde direkt negatif yada pozitif diyemeyiz. Daha ileri araştırma gerekir!



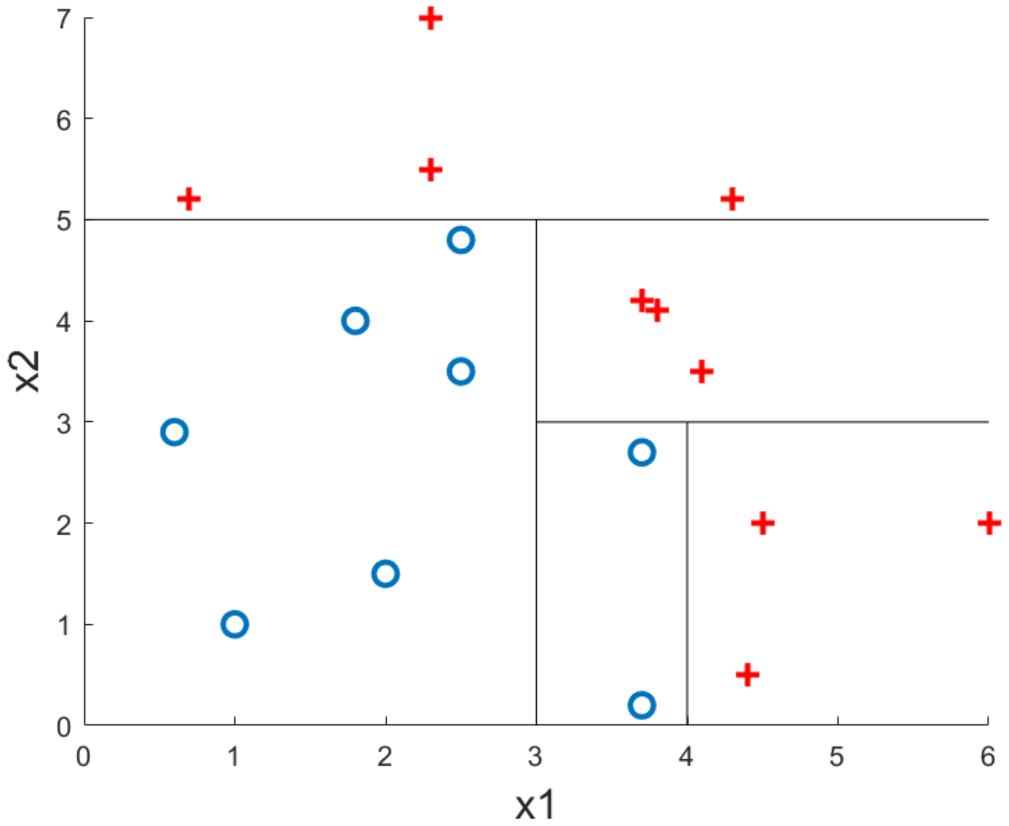


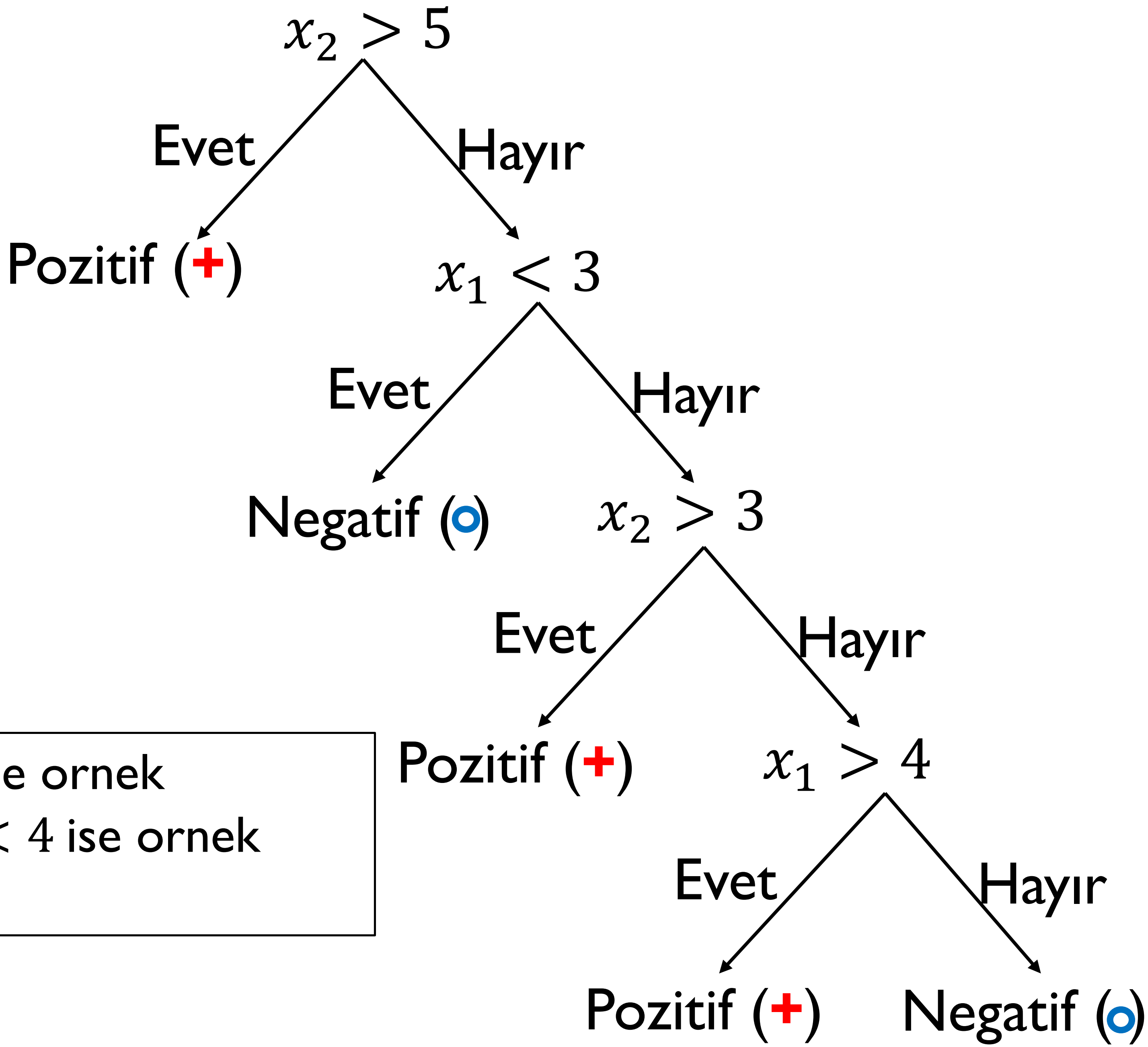
Eger $x_2 < 5$ ve $x_1 < 3$ ise bu ornek negatiftir. olan orneklerde direkt negatif yada pozitif diyemeyiz. $x_2 < 5$ fakat $x_1 > 3$ ise örnek negatif yada pozitif olabilir. Bu noktada daha ileri arastirma gerekir!



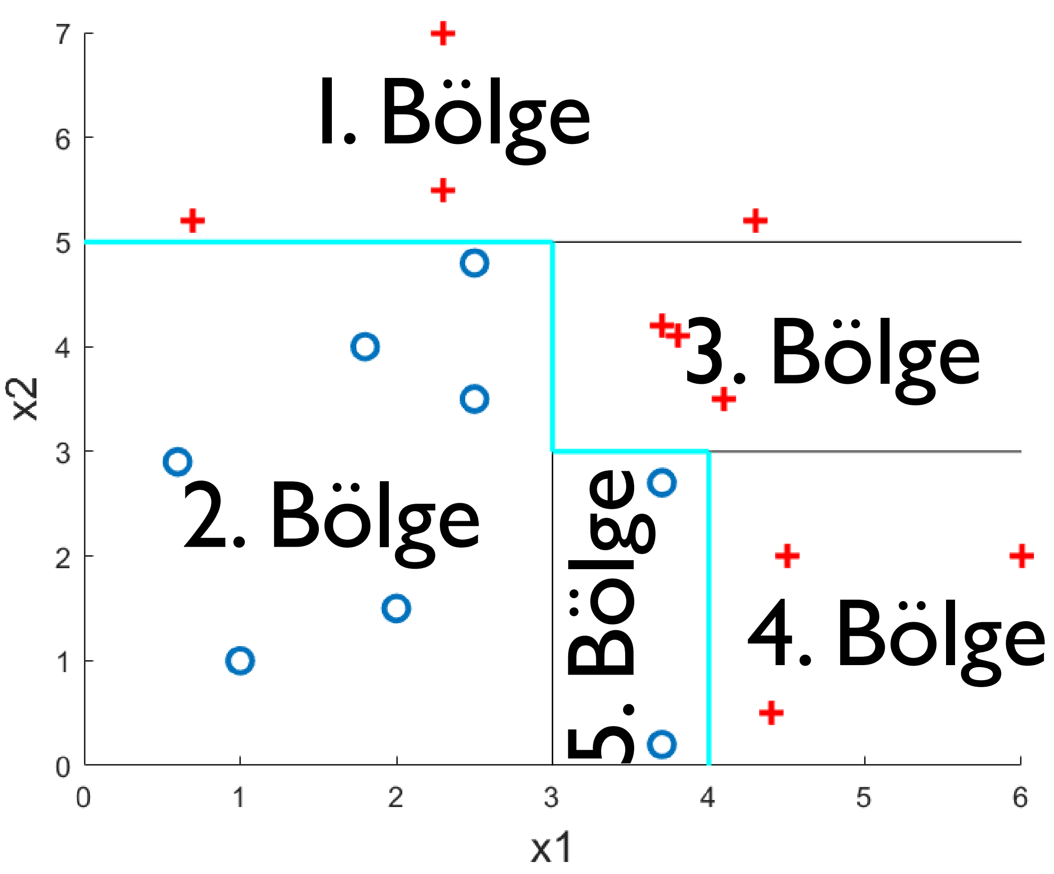


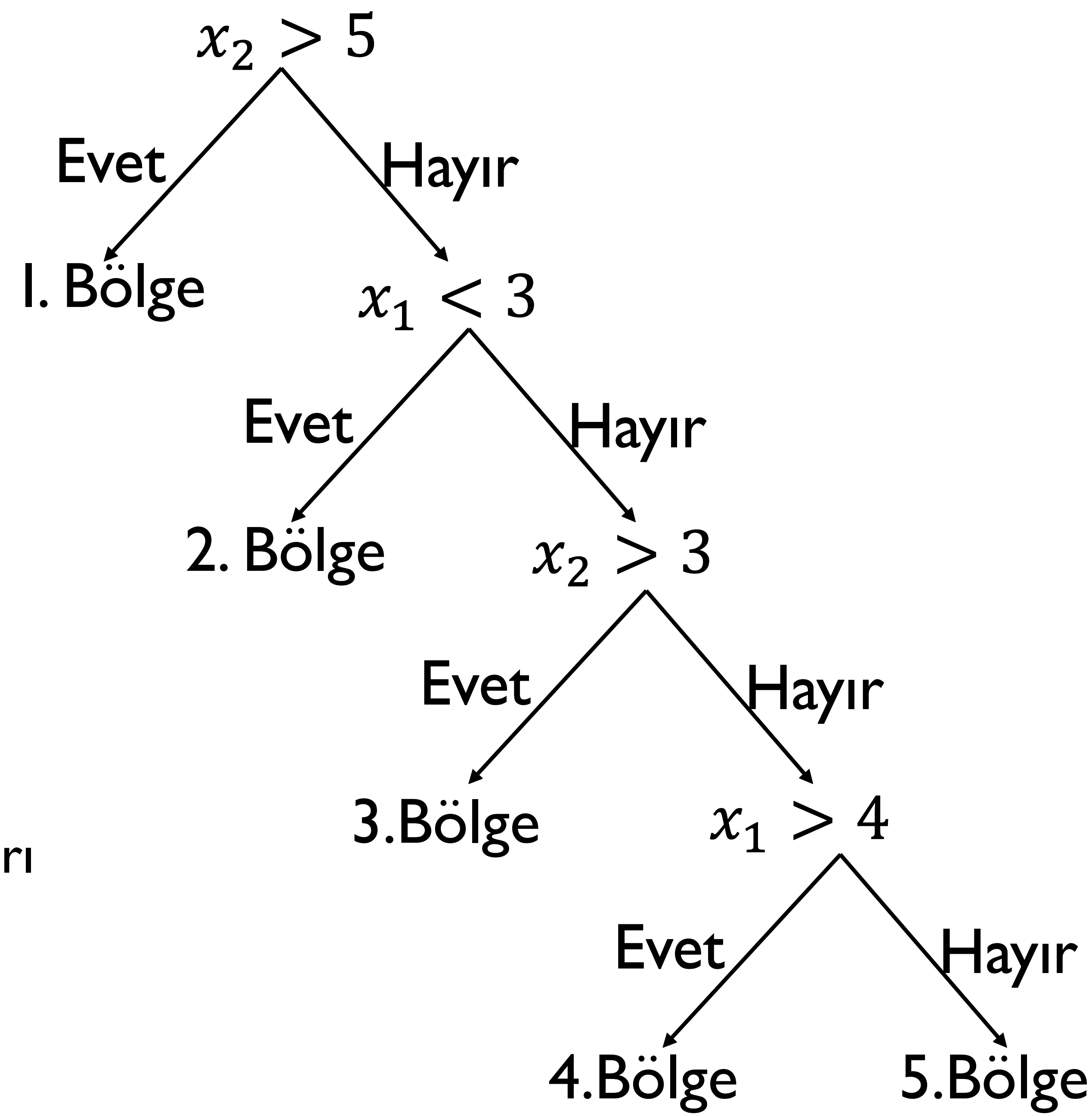
Eger $x_2 < 5, x_1 > 3$ ve $x_2 > 3$ ise bu ornek pozitiftir. Öte yandan örnek $x_2 < 5, x_1 > 3$ ve $x_2 < 3$ ise bu örnek pozitif yada negatiftir diyemeyiz. Bu noktada daha ileri arastirma gerekir!





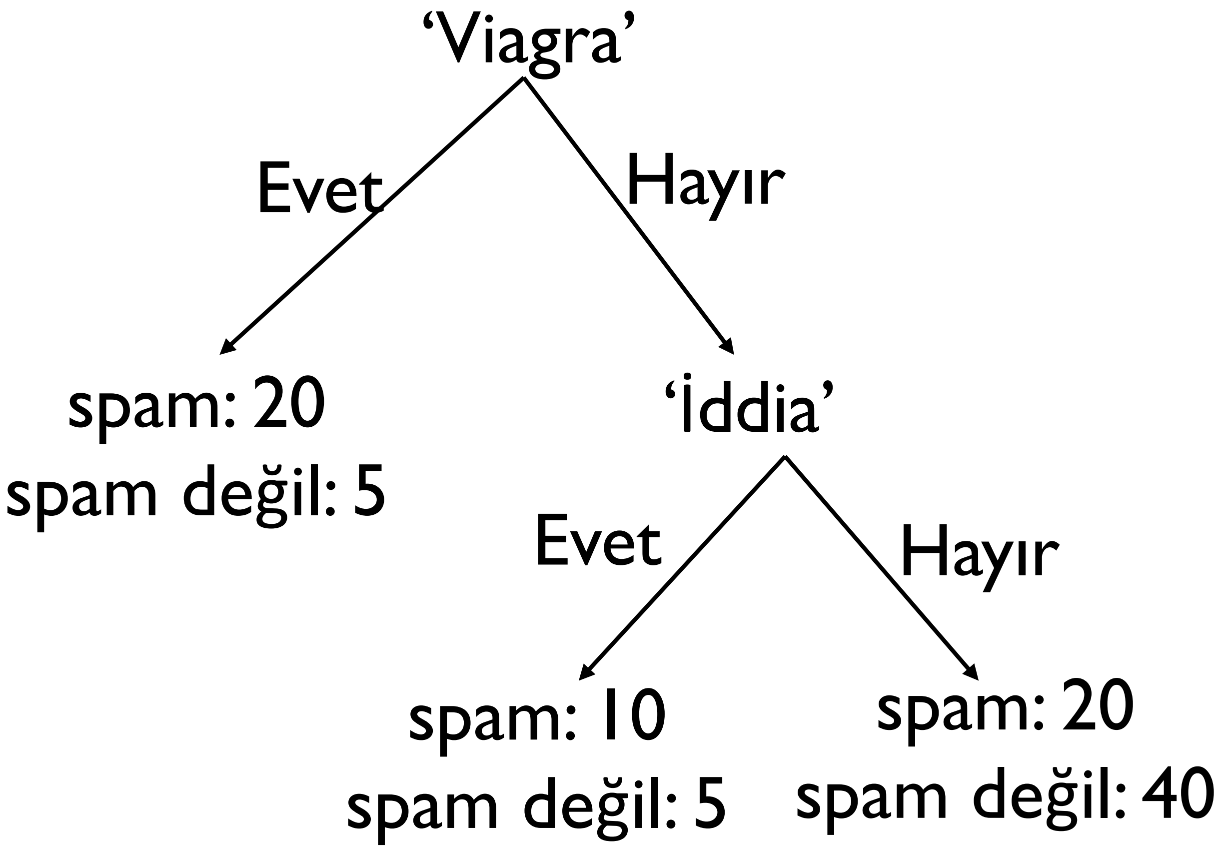
Eger $x_2 < 5$, $x_1 > 3$, $x_2 < 3$ ve $x_1 > 4$ ise ornek pozitiftir; $x_2 < 5$, $x_1 > 3$, $x_2 < 3$ ve $x_1 < 4$ ise ornek negatiftir.



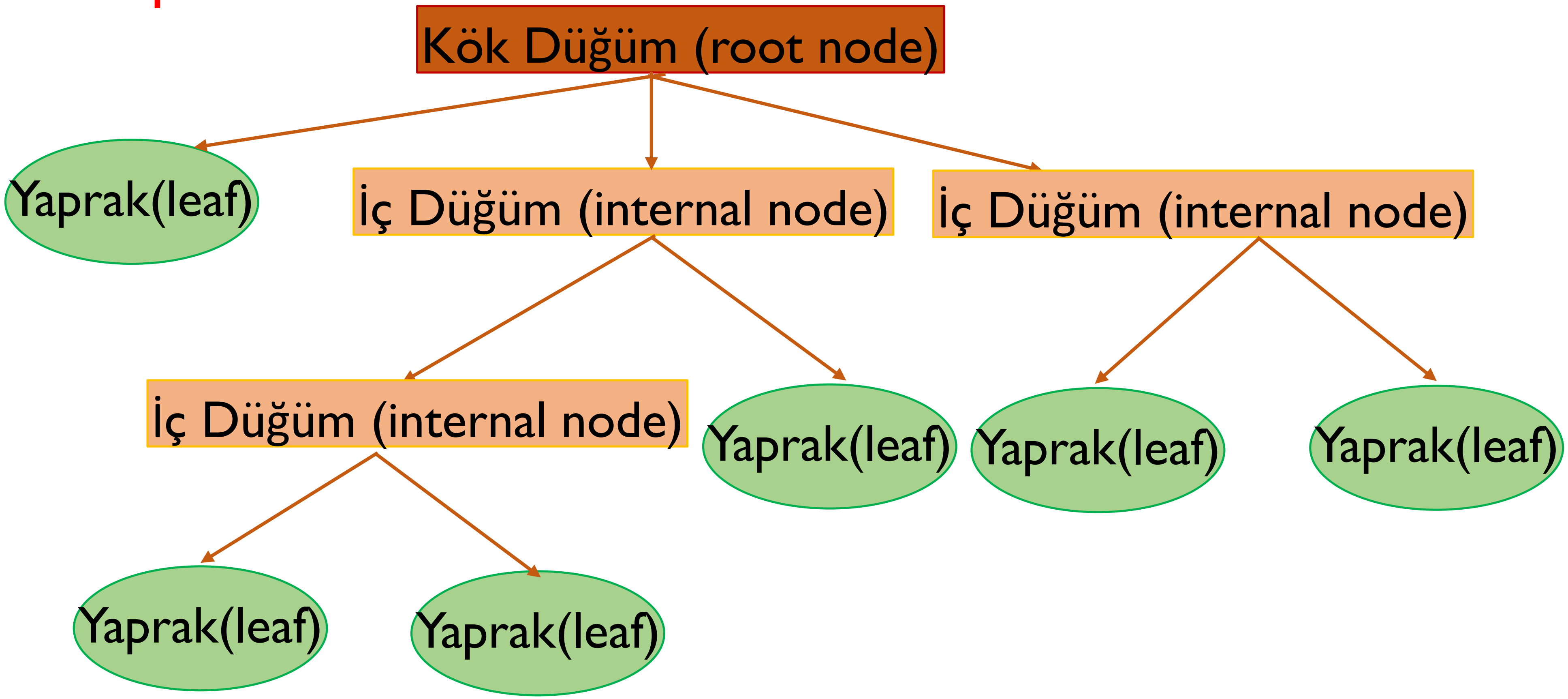


Karar ağacının oluşturduğu karar sınırı
(decision boundary)

Not: Dikkat edilirse oluşturulan her bölgede yalnızca bir sınıftan örnekler var (yalnızca pozitif yada yalnızca negatif örnekler). Bu haliye bu bölgelere *saf* (pure) bölgeler diyeceğiz. Fakat karar agaclariyla cogunlukla tamamen saf bölgeler oluşturmak mümkün değildir.



Karar Ağacının Yapısı



- Bir örneği sınıflandırmaya kök düğümden başlarız.
- İç düğüm, henüz sınıflandırma kararını vermediğimiz noktadır. Bir yaprağa varıncaya kadar ilerleriz. Her iç düğümde bir ozellik test edilir. (iç düğüm de aslinda bir köktür).
- Bir yaprağa vardığımızda siniflandirma işlemi durur. Örnek, bu yapragin ilişkilendirildiği sınıfa eşlenir. Yaprakla ilişkilendirilen sınıf, yaprakta en çok görülen sınıftır.

Karar Ağacı Nasıl İnşaa Edilir?

Bir karar agaci rekürsif (yinelemeli- kendini çagiran) olarak insa edilir.

Eger herhangi bir anda elimizdeki veri seti yeterince homojen ise, yani aşagi yukari ayni sınıfa ait örneklerden oluşuyorsa, agacin bu kisımdan buyumesi durur. Burada yaprak oluşturulur. Yaprak veri setindeki en yaygin sınıf ile etiketlenir.

Bir noktada veri seti yeterince homojen degilse büyüme devam eder. İlk olarak o noktada bir kök oluşturulur. Ve bu noktadaki veri seti icin *en iyi özellik* belirlenir.

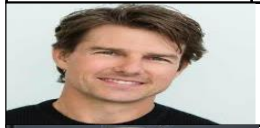
Bu ozelligin aldigi farkli degerlere göre veri seti parçalara ayrilir. Her bir parca icin tekrar agaç büyültme algotirmtasi çagrilarak bir ağaç oluşturulur. Bu agaç köke bağlanır..

```
agacBuyult(V,O) //V: veri seti, O: özellik seti
if homojenMi(V):
   Y yaprağını oluştur;
   Y’yi V’deki en yaygin sınıfla etiketle;
   return Y;
else: // V yeterince homojen degilse
  K kökünü oluştur;
  enIyiOzellik(V,O) ile en iyi özelligi bul: O_enIyi
  K kökününe bulunan en iyi özelligin (O_enIyi) adını ver;
  V’yi O_enIyi’nin her bir değeri icin V_i parçalarına böl;
  for each V_i:
      cocuk=agacBuyult(V_i,O);
      cocuk’u K köküne ekle, baglantiya O_enIyi’nin i. degerini yaz;
  end for;
return K;
```

`homojenMi(V)` bir boolean fonksiyondur: dogru yada yanlisa doner. `V` veri seti yeterince homojen ise dogru'ya degilse yanlis'a döner.

`enIyiOzellik(V,O)`, `V` veri setini en iyi bölen özellige döner. Bu noktayi daha detayli inceleyecegiz.

```
agacBuyult_v2(V,O) //V: veri seti, O: özellik seti
if homojenMi(V):;
   return V'deki en yaygın sınıf;
else: // V yeterince homojen degilse
  enIyiOzellik(V,O) ile en iyi özelligi bul:O_enIyi
  V'yi, O_enIyi'nin her bir değeri icin V_i parçalarına böl;
  for each V_i:
      cocuk_i=agacBuyult(V_i,O);
  end for;
return cocuklari cocuk_i olan agac;
```

Kişi	Ev sahibi	Evlilik durumu	Yıllık gelir	Krediyi ödemiş mi?
	Evet	Bekar	125K	Hayır
	Hayır	Evli	100K	Hayır
	Hayır	Bekar	70K	Hayır
	Evet	Evli	120K	Hayır
	Hayır	Bekar	95K	Evet
	Hayır	Evli	60K	Hayır
	Evet	Bekar	220K	Hayır
	Hayır	Bekar	85K	Evet
	Hayır	Evli	75K	Hayır
	Hayır	Bekar	90K	Evet

V={, , , , , , , , , }

O={ev sahibi, evlilik durumu, yıllık gelir}

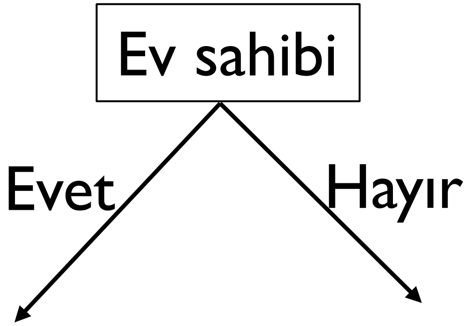
`agacBuyult(V,O)` algoritmasina başlayalım...

`homojenMi(V)=hayır.` (krediyi odemeyen 7 kişi, krediyi ödeyen 3 kişi)

`enIyiOzellik(V,O)=`O_{enIyi}= Ev sahibi

'Ev sahibi' (ilk) kök olur.

Ev sahibi'nin değerleri: Evet, Hayır. O halde Ev sahibi kökünden 'Evet' ve ' Hayır' dalları çıkar.



Ev sahibi özelliği veri setini ikiye ayırır. Ev sahibi olanlar: V_{Evet}, ev sahibi olmayanlar: $V_{\text{Hayır}}$

$V_{\text{Evet}} = \{$ , ,  $\}$

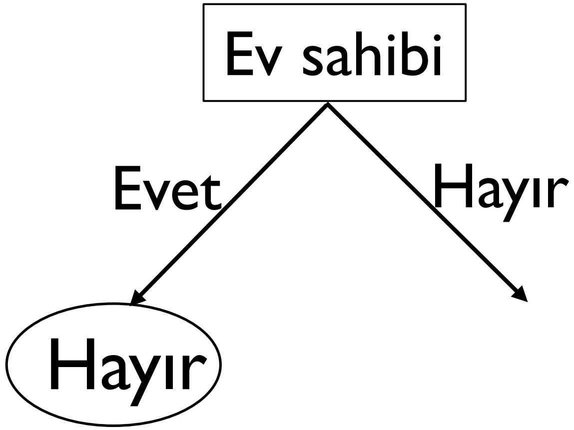
$V_{\text{Hayır}} = \{$ , , , , , ,  $\}$

Bu elde edilen iki veri seti icin **`agacBuyult(V,0)`** algoritmasini çagiralim.

`agacBuyult(`V_{Evet}**`,0):`**

`homojenMi(`V_{Evet}`)=evet.` (V_{Evet}'tekilerin tamamı aldiği krediyi ödemiş).

Şu halde bir yaprak oluşturalım. Oluşan bu yaprağin sınıfı 'Hayır'tir; çunku bu yapraktakiler aldiği krediyi ödememistir. Ağacin buyumesi bu noktada durur.



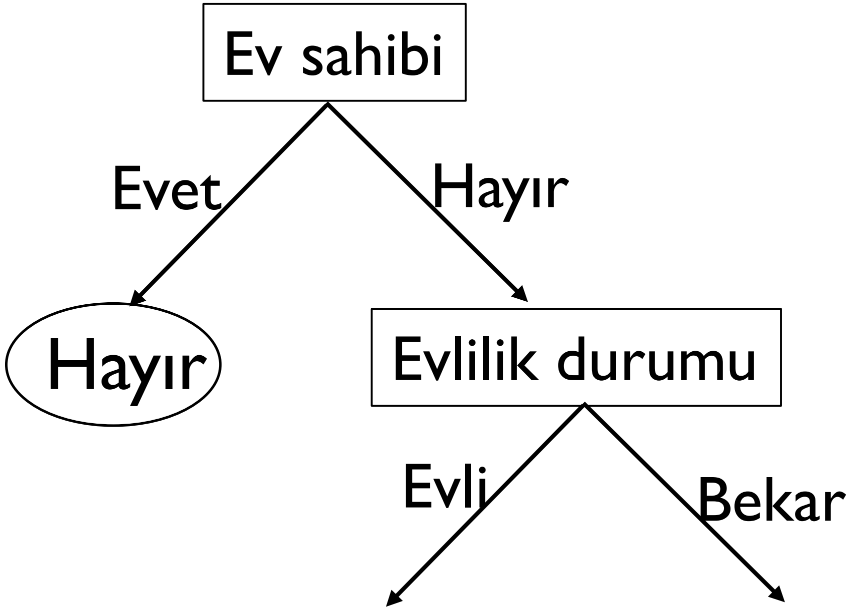
`agacBuyult(`$V_{Hayır}$`,O):`

`homojenMi(`$V_{Hayır}$`)=Hayır.` ($V_{Hayır}$'dakilerin bazıları aldiği krediyi ödemiş, bazilari ödememiş).

`enIyiOzellik(`$V_{Hayır}$`,O)=`O_{enIyi}= Evlilik durumu

'Evlilik durumu' kök olur.

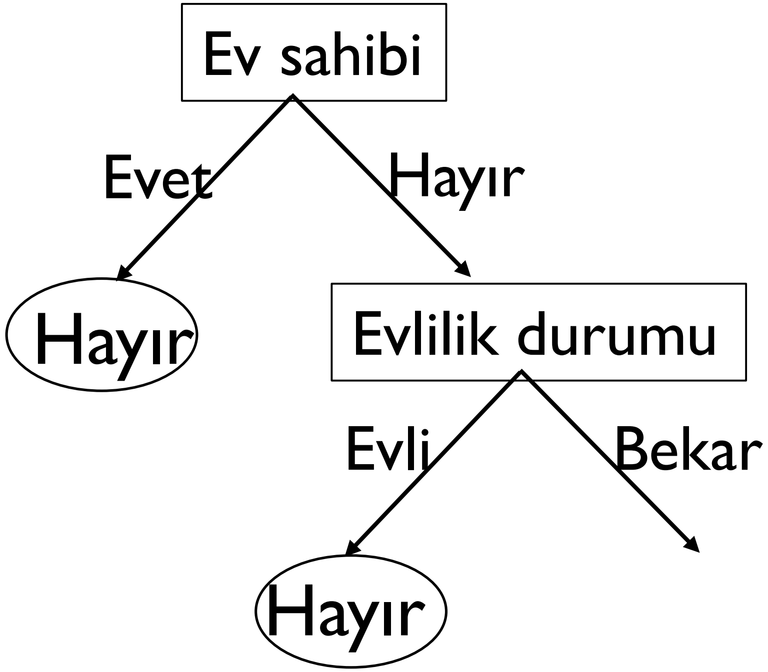
Evlilik durumu'nun değerleri: Evli, Bekar. O halde Evlilik durumu kökünden 'Evli' ve ' Bekar' dalları çıkar.



$V_{\text{Hayır,Evli}}$ ={  , ,  }

`agacBuyult(`$V_{\text{Hayır,Evli}}$**`,0):`**

`homojenMi(`$V_{\text{Hayır,Evli}}$`)=evet.` O halde bir yaprak oluşturalım. Oluşan bu yaprağin sınıfı 'Hayır'dır; çunku ev sahibi olmayip evli olanlarin aldiği krediyi ödememistir.



$V_{\text{Hayır,Bekar}}$ ={, , , }

agacBuyult($V_{\text{Hayır,Bekar}}$**,0):**

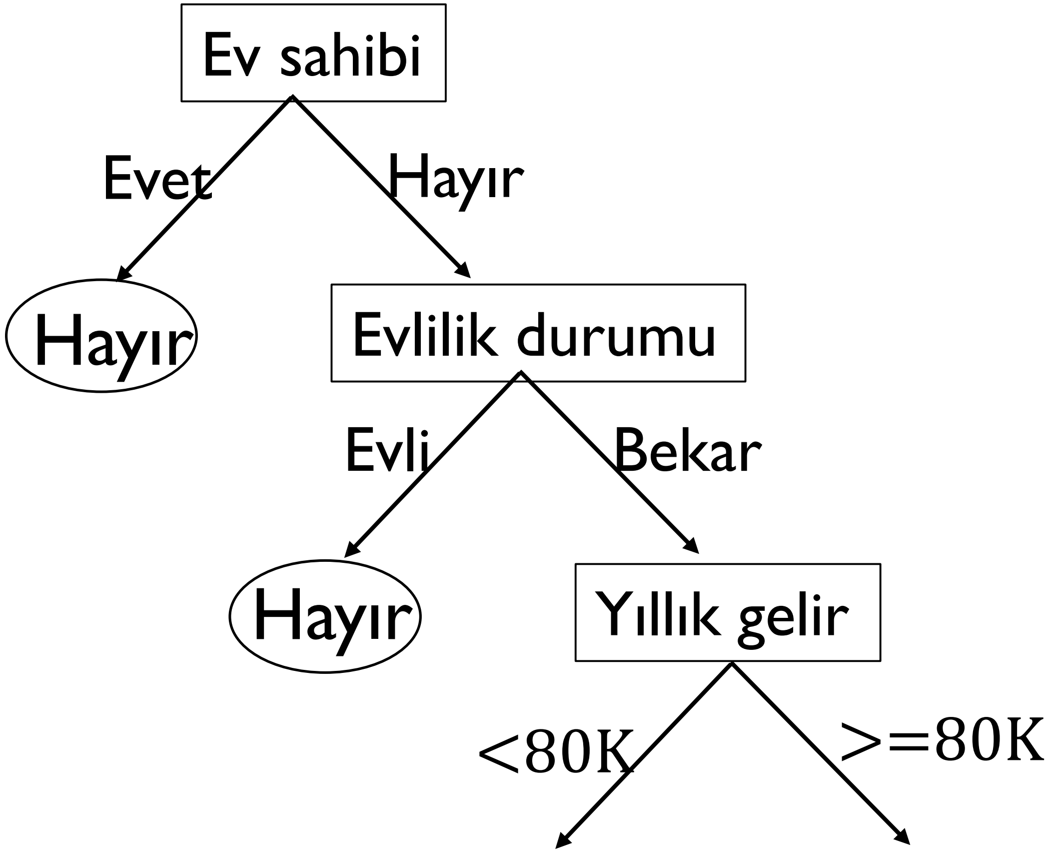
homojenMi($V_{\text{Hayır,Bekar}}$)=hayır. (Ev sahibi olmayip bekar olanlarin bir kısmı (3 kişi) aldiği krediyi odemis, bazisi (1 kişi) ödememistir).

enIyiOzellik($V_{\text{Hayır,Bekar}}$, O)=O_{enIyi}=Yıllık gelir.

'Yıllık gelir' kök olur.

Diğer iki özelligin aksine Yıllık gelir kategorik degil, sayisal bir ozelliktir. Sonsuz deger alir. Teorik olarak alacagi her deger icin bir dal çikarmak gerekir. Bu mumkun olmadigindan bir eşik değeri buluruz. Bu eşik degerinin nasil bulunacagini daha sonra anlaticagiz.

Bu eşik degeri 80K olsun. Yillik gerliri 80K dan az olanlar icin bir dal, 80K dan fazla olanlar icin baska bir dal olusturacagiz.



$V_{\text{Hayır,Bekar,80K\_az}}$={  }

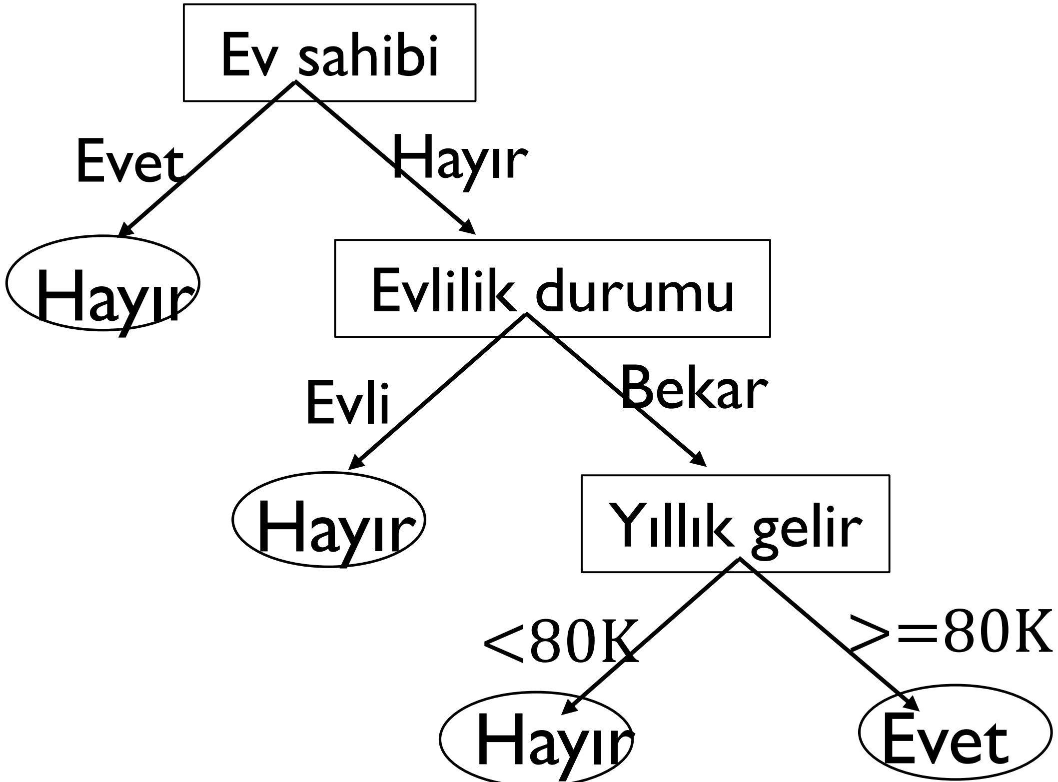
$V_{\text{Hayır,Bekar,80K\_fazla}}$={  ,  ,  }

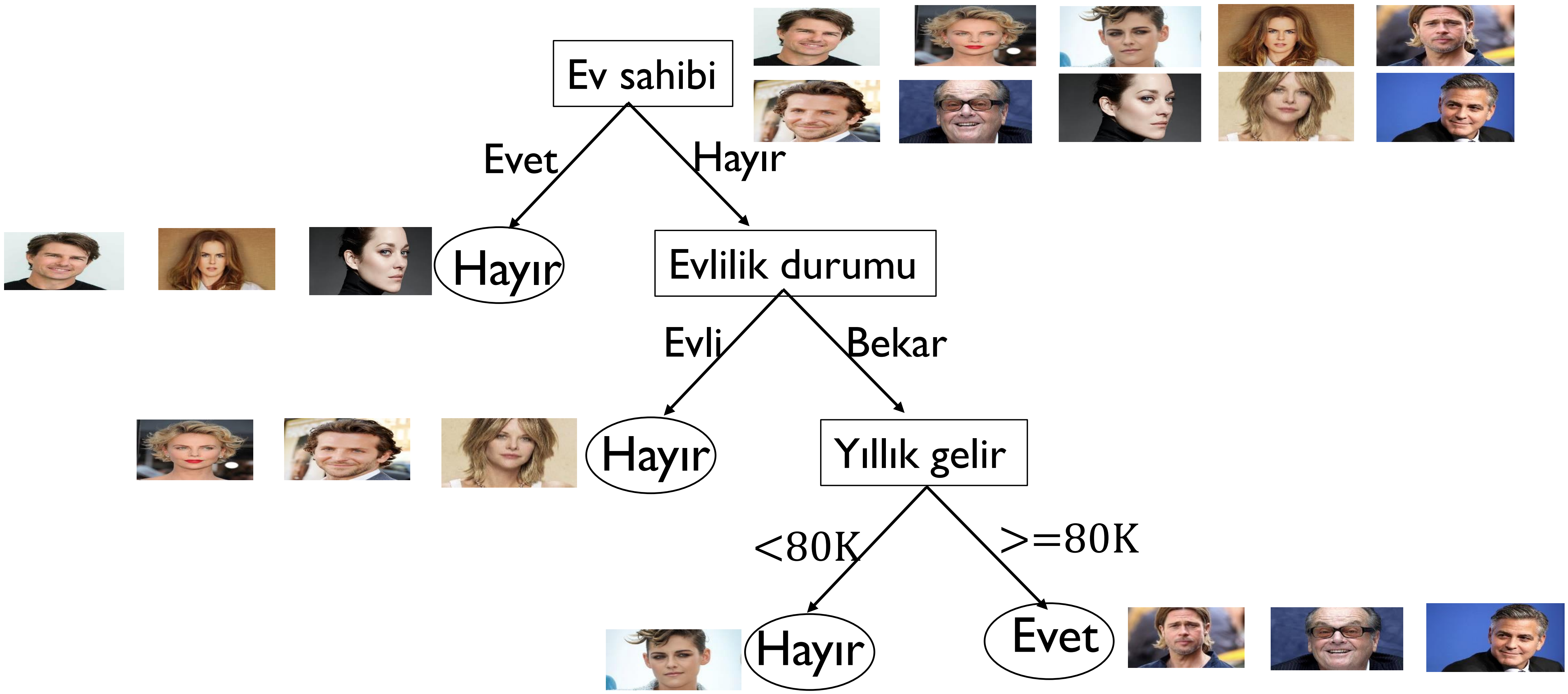
agacBuyult($V_{\text{Hayır,Bekar,80K\_az}}$,0) :

homojenMi($V_{\text{Hayır,Bekar,80K\_az}}$)=evet. Yaprak olusturalim. Oluşan bu yaprağin sınıfı 'Hayır'dır; çunku ev sahibi olmayip, bekar ve geliri 80K dan az olanlar aldigi krediyi odememistir.

agacBuyult($V_{\text{Hayır,Bekar,80K\_fazla}}$,0) :

homojenMi($V_{\text{Hayır,Bekar,80K\_fazla}}$)=evet. Yaprak olusturalim. Oluşan bu yaprağin sınıfı 'Evettir'dır; çunku ev sahibi olmayip, bekar ve geliri 80K dan fazla olanlar aldigi krediyi odemistir.




Ev sahibi
Evet
Hayır
Hayır
Evlilik durumu
Evli
Bekar
Hayır
Yıllık gelir
<80K
>=80K
Hayır
Evet

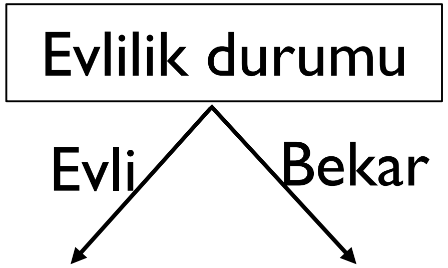
Özelliklerin Bölünmesi

Özellikler

Kategorik

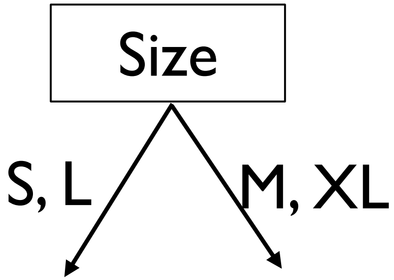
Sembolik

Her bir değer için ayrı ayrı bölünür



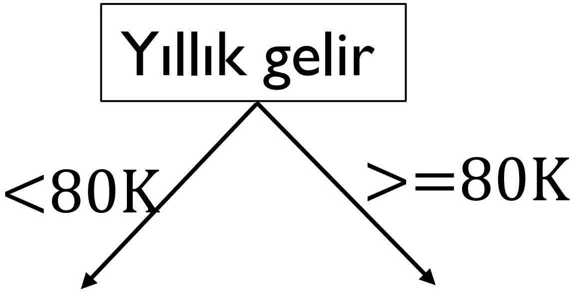
Sıralı

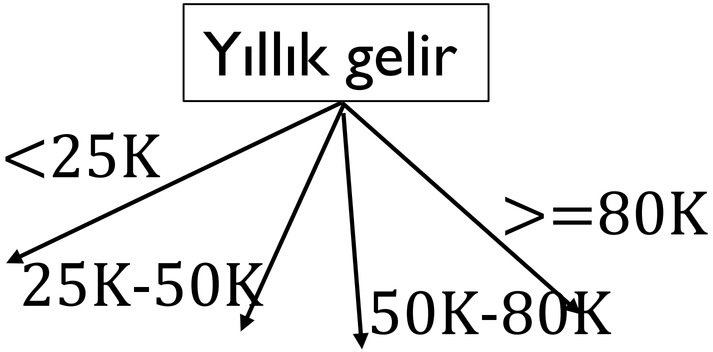
Ya her bir değer için ayrı ayrı bölünür; yada bazı değerler gruplandıktan sonra gruplar arasında bölünme olur



Sayısal

Bir yada birkaç eşik değeri belirlenir. Bu eşiklere göre bölünme yapılır

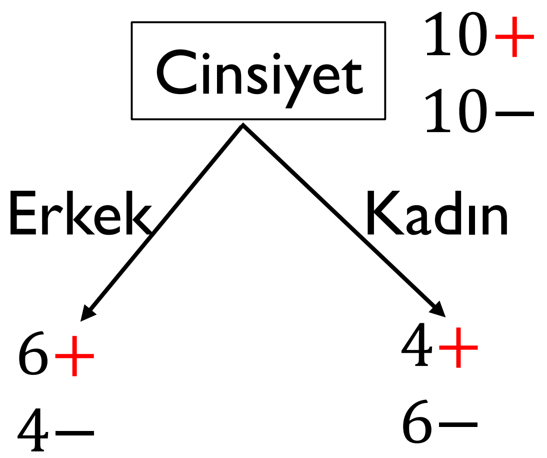


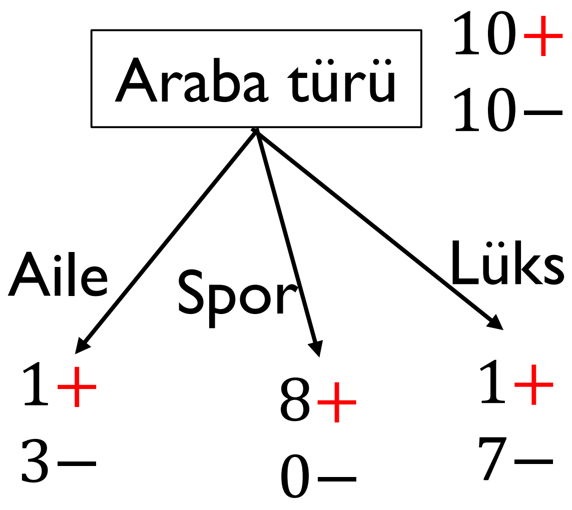


Karar Ağacı Oluştururken En Iyi Özellik Nasıl Bulunur?

Karar agacinda bir kök için, bu kökü parçalara ayıracak özellik seçerken, her bir parçada sınıf dagilimi en saf/en homojen olacak şekilde özellik secilimi yapilir. Her bir parçada süpriz/belirsizlik miktari en düşük olmaldir.

ör. Bir karar agacinda belirli bir kokte elimizde 20 tane ornek olsun. Bu orneklerin 10 tanesi pozitif sınıfa, 10 tanesi negatif sınıfa ait olsun. Bu kök homojen olmadigindan bunu homojen olacak parçalara ayırırız. Diyelimki elimizde iki özellik olsun: cinsiyet ve araba türü.





Araba türü'nün oluşturduğu parçalar daha homojen; hangi sınıfın olacagi belirsizligi daha düşük!

Süprizin (Belirsizliğin) Ölçülmesi

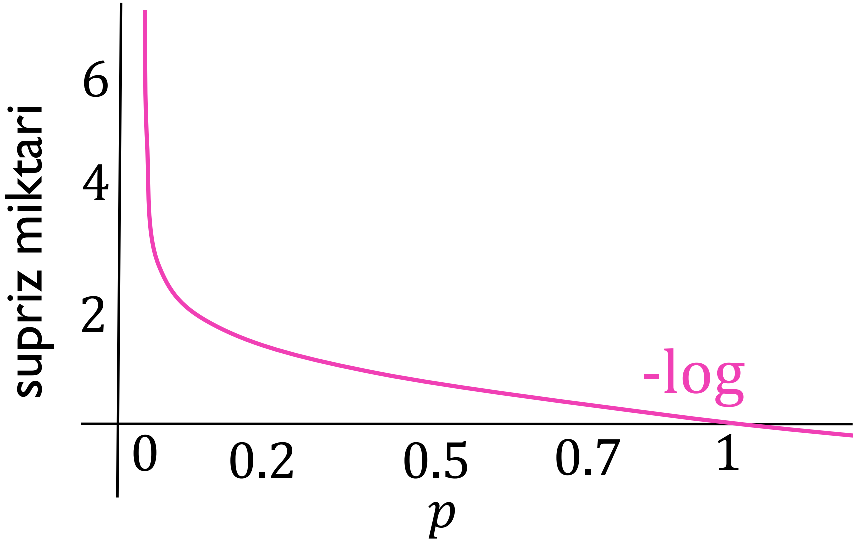
Information theory'de p olasiligina sahip bir olay gerçekleştigindeki şaşkınlığımız $-\log p$ ile ölçülür. Bir başka ifadeyle $-\log p$, p olasiligina sahip bir olayin gerceklesmesinin bizim icin ne kadar *supriz* oldugunu gosterir.

ör. Diyelimki elimizde hileli bir bozuk para var. Bu parada tura gelmesi olasılığı $P(tura) = 0.2$, yazı gelmesi olasılığı $P(yazı) = 0.8$ olsun.

Tura gelmesindeki supriz miktari: $-log0.2 = 1.6$

Yazi gelmesindeki supriz miktari: $-log0.8 = 0.2$

Atılan bu parada tura gelmesi daha buyuk suprizdir (şaşkınlık miktarimiz daha fazladir); öte yandan yazı gelmesi pek de süprız degildir.



$p = 0$ iken (yani olayin gerceklesme olasiligi yok iken), eger olay gerceklesirse supriz sonsuz (infinity) olur.

$p = 1$ iken (yani olayin gerceklesmesi kesin iken), olay gerceklesirse supriz 0 olur.

Entropy (Ortalama Supriz)

p_i'ler K tane olayin gerceklesme olasiligi olsun ($\sum_{i=1}^{K} p_i = 1$). Bu durumda entropy:

$$\boxed{\sum_{i=1}^{K} p_i \cdot -\log p_i}$$

her bir suprizi, bu suprizin görulme olasligi ile carpiyoruz

Karar ağaçlarına dönersek;

Entropy'yi homojen olmamanın ölçusu olarak kullanacagiz.

(Entropy safsızlığın/belirsizliğin ölçusudur).

Yani bir düğümde entropy yüksekse, burada baskın bir sınıf yoktur. Sınıflardan benzer sayılarda örnekler vardir (örnegin 10 pozitif, 10 negatif ornek).

p_i, bir V kumesinde i. sınıfın görülme olasılığı olsun ve bu düğümde toplam K tane sınıf olsun. Bu durumda V kumesindeki entropy $H(\mathrm{V}) = \sum_{i=1}^{K} p_i \cdot -\log p_i$

Diyelim ki bir O özelliginin n farkli degeri olsun: $O_1, 0_2, \ldots, O_n$

n deger V kumesini n parçaya ayirir: $V_1, V_2, \ldots, V_n$

V düğümü bir O özelligine gore n parçaya ayrilsin.

Bu parçalarin herbirindeki entropiyi hesaplarız: $H(V_1), H(V_2), \ldots, H(V_n)$.

Daha sonra bulunan entropileri $P(O_1), P(O_2), \ldots, P(O_n)$ olasiliklasiri ile carparak toplariz:

V kumesinin bir O özelligi ile n parçaya bolunmesi ile ortaya cikan entropy:

$$\sum_{i=1}^{n} P(O_i)H(V_i)$$

V kumesinin O ile bolunmeden onceki entropysi (belirisizligi) $H(V)$ idi. Yukarida hesaplanan yeni entropi ile arasindaki fark *bölünme ile oluşan kazanç miktari*ni verir. Bu da şu şekilde hesaplanir:

$$Kazanç\, miktari\ (O) = H(V) - \sum_{i=1}^{n} P(O_i)H(V_i)$$

En iyi özellik maksimum kazanç miktari saglayan özelliktir; yani belirsizligi en cok dusuren ozelliktir.