Karar ağaçları

Decision Trees Nedir?

- Decision Tree genellikle sınıflandırma problemleri için kullanılan bir gözetimli öğrenme algoritmasıdır. Regresyon içinde kullanılabilir. Yüksek doğruluk, kararlılık ve yorumlanma kolaylığına sahiptir. Doğrusal olmayan ilişkileride oldukça iyi tespit edebilir.

Makine Öğrenimi Karar Ağaçları Ne İçin Kullanılır?

- Bu model çok yönlüdür, dolayısıyla karar ağacı algoritması makine öğreniminde birçok uygulama alanı bulmaktadır:
- Veri madenciliği
- Veri bilimi
- İstatistikler
- Bu tekniğin amacı, çeşitli girdi değişkenlerine dayalı olarak bir hedef değişkenin değerini tahmin eden bir model oluşturmaktır. Bu teknik şu gibi soruları cevaplamak için kullanılır: X yapılmalı mı? A'yı mı yoksa B'yi mi seçmeliyiz?

1. Karar analizi

- Karar verme sürecinde kullanılır. Karar ağacı makine öğrenimi modeli, karar sürecini ve sonucunu açıkça sunmak için veri görselleştirmede kullanılabilir. Bu, karar ağaçlarının en büyük avantajlarından biri, işlerinde makine öğrenimi ile ilgilenmeyen biri için bile anlaşılabilir ve kolay olmasıdır.

2. Sınıflandırma

- Sınıflandırma, bir sınıf değerini tahmin etmek veya açıklamak için kullanılan bir makine öğrenimi tekniğidir. Sınıflandırma algoritmaları ile bir veya daha fazla girdiye dayalı olarak bir olayın meydana gelme olasılığını değerlendirebilirsiniz. Karar ağacı makine öğrenimi modeli tam olarak bununla ilgilidir. Örneğin, e-posta'ları spam ve spam olmayan olarak ikiye ayırmak için kullanılabilir. Algoritmanız bir dizi soru sorar ve yanıtlara göre belirli bir e-posta'nın spam olup olmadığına karar verir.

3. Regresyon

- Denetimli makine öğrenimi tekniklerinden biridir. Regresyon, bir dizi önceki veriye dayanarak belirli bir değeri tahmin etmenize (veya açıklamanıza) yardımcı olur. Bir çalışanın maaşı, hastalığın yayılması veya mülk değeri gibi konuları tahmin etmek için regresyonu kullanabilirsiniz. Bu durumda, buna regresyon ağacı denir.
- Gerçek yaşam koşullarında bu model insanlar tarafından yaygın olarak kullanılmaktadır. Arkadaşlarınızla buluşup buluşmayacağınızı ya da televizyon izleyip izlemeyeceğinizi her düşündüğünüzde, sadece zihninizde olsa bile karar ağacı tekniğini kullanırsınız. Makine öğreniminin çeşitli yönlerinde bu kadar yaygın olmasının nedeni budur. Bu, insanların öğrenme şeklini taklit etmeye çalışan bir teknolojidir ve insanlar farkında olmasalar bile bu tekniği kullanır.

Karar ağaçlarında terimler

- Karar ağaçlarının ilk hücrelerine **kök** (root veya root node) denir. Her bir gözlem kökteki koşula göre “Evet” veya “Hayır” olarak sınıflandırılır.
- Kök hücrelerinin altında **düğümler** (interval nodes veya nodes) bulunur. Her bir gözlem düğümler yardımıyla sınıflandırılır. Düğüm sayısı arttıkça modelin karmaşıklığı da artar.
- Karar ağacının en altında **yapraklar** (leaf nodes veya leaves) bulunur. Yapraklar, bize sonucu verir.

Karar ağacı kullanılırken yapılan bazı varsayımlar aşağıdadır:

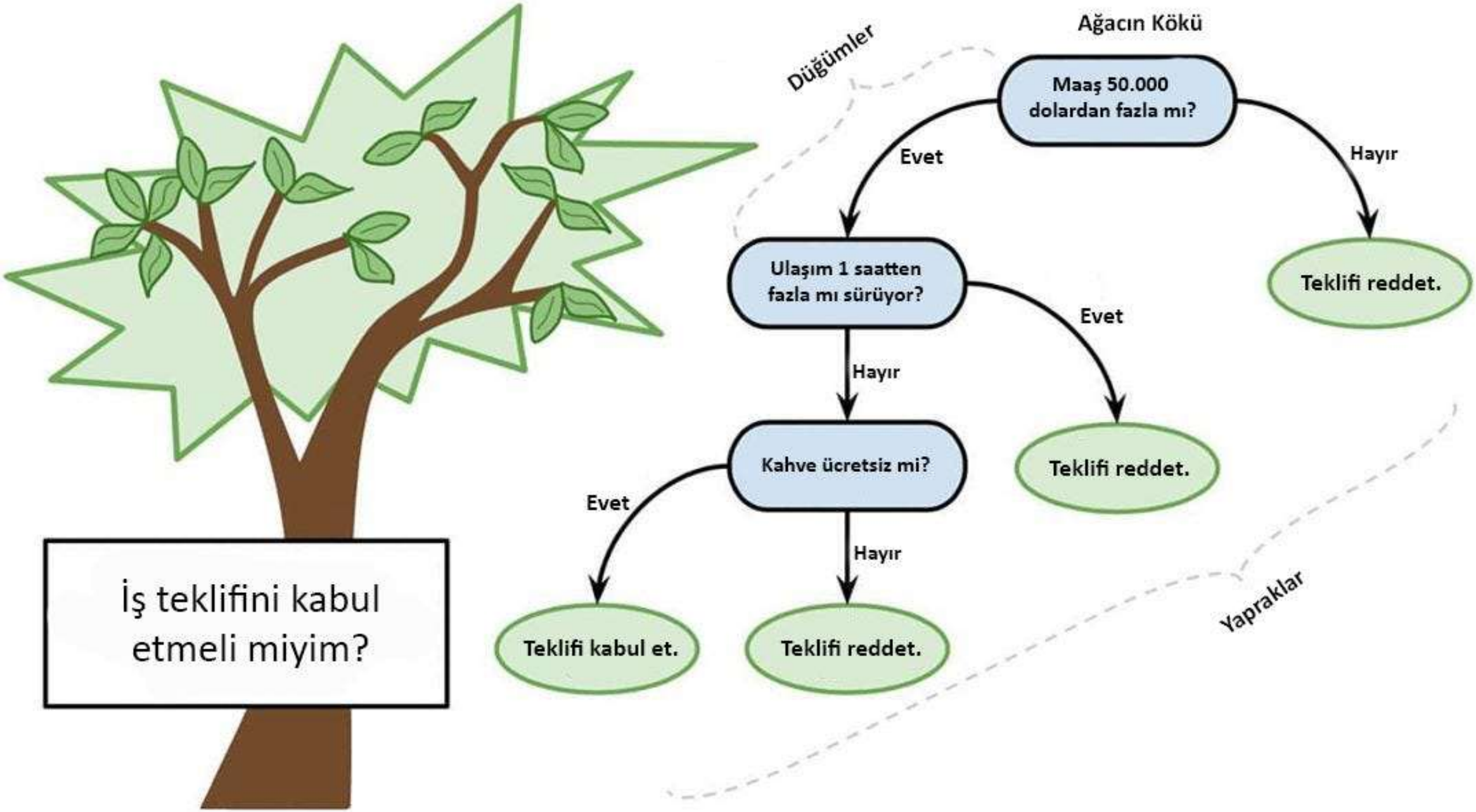
- Başlangıçta, tüm eğitim seti kök olarak kabul edilir.
- Özellik değerlerinin kategorik olması tercih edilir. Değerler sürekli ise, model oluşturmadan önce ayrıklaştırılırlar.
- Öznitelik değerleri temelinde, kayıtlar özyinelemeli olarak dağıtılır.
- Özniteliklerı kök veya dahili düğüm olarak sıralamak için istatistiksel yöntemler kullanılır.

Karar Ağacı Algoritması:

- 1) T öğrenme kümesini oluşturulur.
- 2) T kümesindeki örnekleri en iyi ayıran nitelikler belirlenir.
- 3) Seçilen nitelik ile ağacın düğümleri oluşturulur ve herbir düğümde alt düğümler veya ağacın yapraklarını oluşturulur. Alt düğümlere ait alt veri kümesinin örneklerini belirlenir

Karar Ağacı Algoritması:

- 4) 3. adımda oluşturulan her alt veri kümesi için ;
  - Örneklerin hepsi aynı sınıfa aitse
  - Örnekleri bölecek nitelik kalmamışsa
  - Kalan niteliklerin değerini taşıyan örnek yoksa işlemi sonlandır. Diğer durumda alt veri kümesini ayırmak için 2. adımdan devam edilir.


İş teklifini kabul etmeli miyim?
Düğümler
Ağacın Kökü
Maaş 50.000 dolardan fazla mı?
Evet
Hayır
Teklifi reddet.
Ulaşım 1 saatten fazla mı sürüyor?
Evet
Hayır
Teklifi reddet.
Kahve ücretsiz mi?
Evet
Hayır
Teklifi kabul et.
Teklifi reddet.
Yapraklar

Karar Ağaçlarının Oluşturulması

- Karar ağaçları, bir dizi kural kullanılarak veri setinden oluşturulur. Bu kurallar genellikle verinin belirli özelliklerine dayalı olarak bölünmesini içerir. Ağaç oluşturma sürecinde kullanılan yaygın yöntemler:
- **ID3 (Iterative Dichotomiser 3):** Entropi ve bilgi kazancı kullanarak veri setini böler.
- **C4.5:** ID3 algoritmasının geliştirilmiş versiyonudur ve kategorik verilerle daha iyi çalışır.
- **CART (Classification and Regression Trees):** Gini indeksi veya varyans azaltma yöntemlerini kullanarak veri setini böler.

Entropi

- **Entropi:** Bir veri kümesinin düzensizliğini veya belirsizliğini ölçer. Yüksek entropi, verinin daha karışık olduğunu gösterir.

$$Entropy H(X) = -\left[\left(\frac{3}{8}\right) log_2 \frac{3}{8} + \left(\frac{5}{8}\right) log_2 \frac{5}{8}\right]$$

```
= -[0.375 * (-1.415) + 0.625 * (-0.678)]
=-(-0.53-0.424)
= 0.954
```

Bilgi Kazancı

- **Bilgi Kazancı:** Bir özelliğin veriyi ne kadar iyi böldüğünü ölçer. Yüksek bilgi kazancı, daha iyi bir bölünme anlamına gelir.

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

Ezber (Overfitting: Aşırı Uyum):

- Tüm makine öğrenmesi yöntemlerinde verinin ana hatlarının modellenmesi esas alındığı için öğrenme modelinde ezberden (overfitting) kaçınılmalıdır.
- Tüm karar ağaçları önlem alınmazsa ezber yapar. Bu yüzden ağaç oluşturulurken veya oluşturulduktan sonra budama yapılmalıdır.

Ağaç Budama:

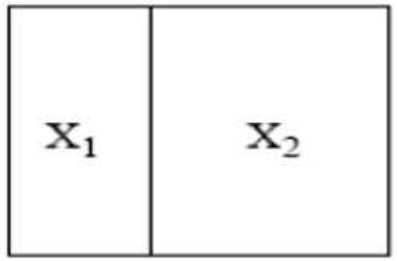
- Budama, sınıflandırmaya katkısı olmayan bölümlerin karar ağacından çıkarılması islemidir. Bu sayede karar ağacı hem sade hem de anlaşılabilir hale gelir. İki çesit budama yöntemi vardır; – Ön budama – Sonradan budama Ön budama islemi ağaç oluşturulurken yapılır. Bölünen nitelikler, değerleri belli bir esik değerinin (hata toleransının) üstünde değilse o noktada ağaç bölümleme islemi durdurulur ve o an elde bulunan kümedeki baskın sınıf etiketi, yaprak olarak olusturulur. Sonradan Budama: Sonradan budama islemi ağaç olusturulduktan sonra devreye girer. Alt ağaçları silerek yaprak olusturma, alt ağaçları yükseltme, dal kesme seklinde yapılabilir.

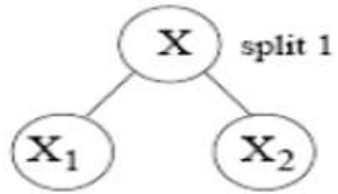
Budama işleminde ne zaman duracağınıza karar vermenin birkaç yolu vardır:

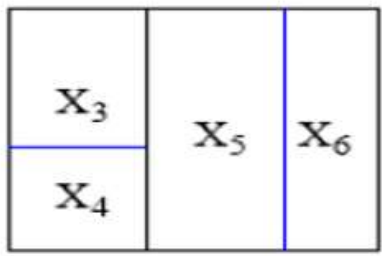
- Tüm terminal düğümleri saf olana kadar devam edilir.
- Her bir terminal düğümündeki veri sayısı belirli bir eşikten, örneğin 5'ten, hatta 1'den büyük olmayana kadar devam edilir.
- Ağaç yeterince büyük olduğu sürece, ilk ağacın boyutu kritik değildir.

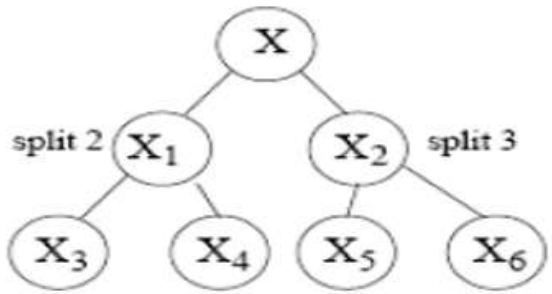
- Buradaki anahtar, ilk ağacı yeniden budamadan önce yeterince büyük yapmaktır!

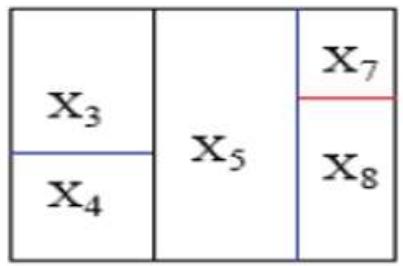
Sınıflandırma Ağaçları:

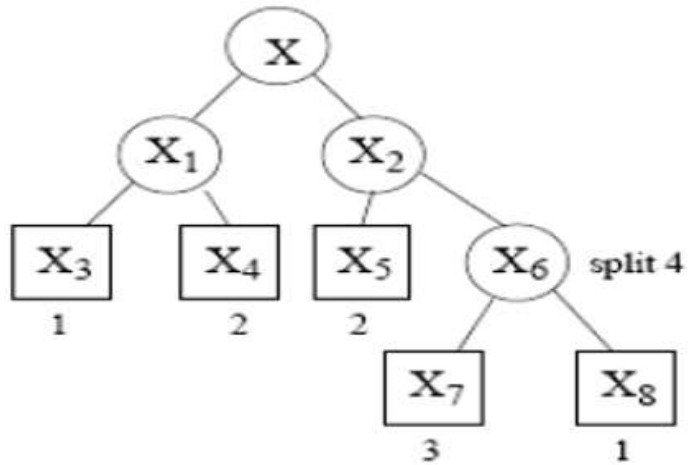


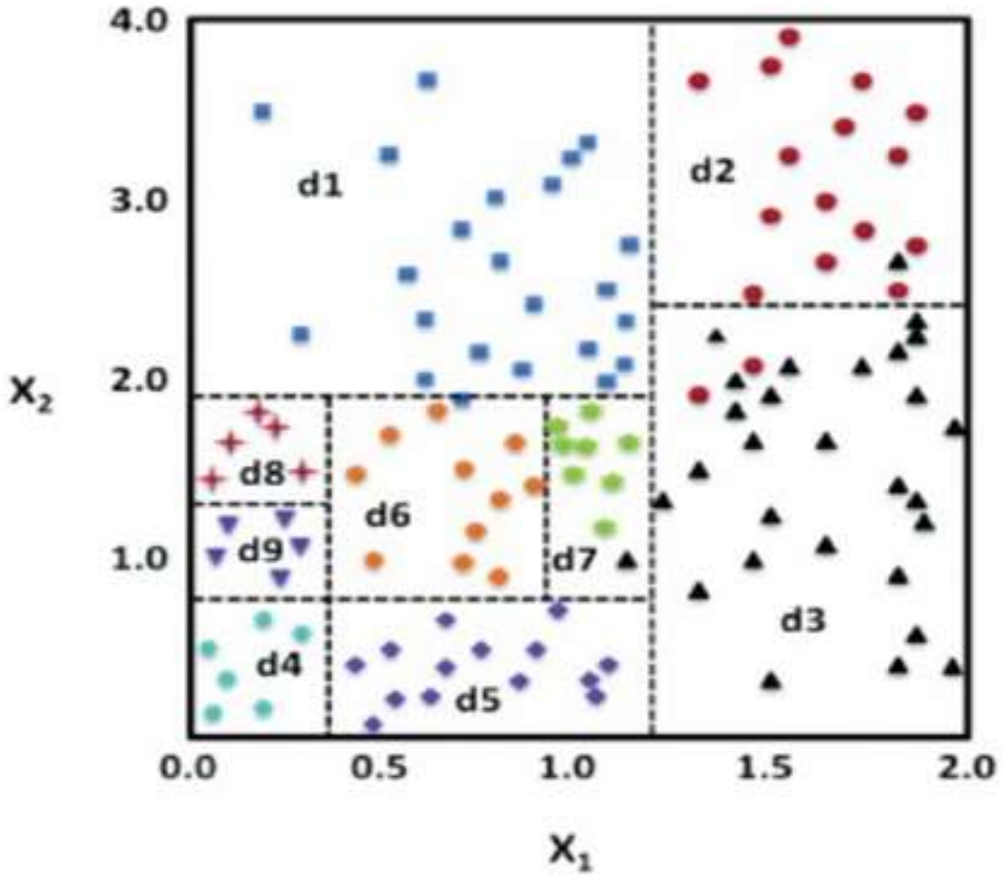




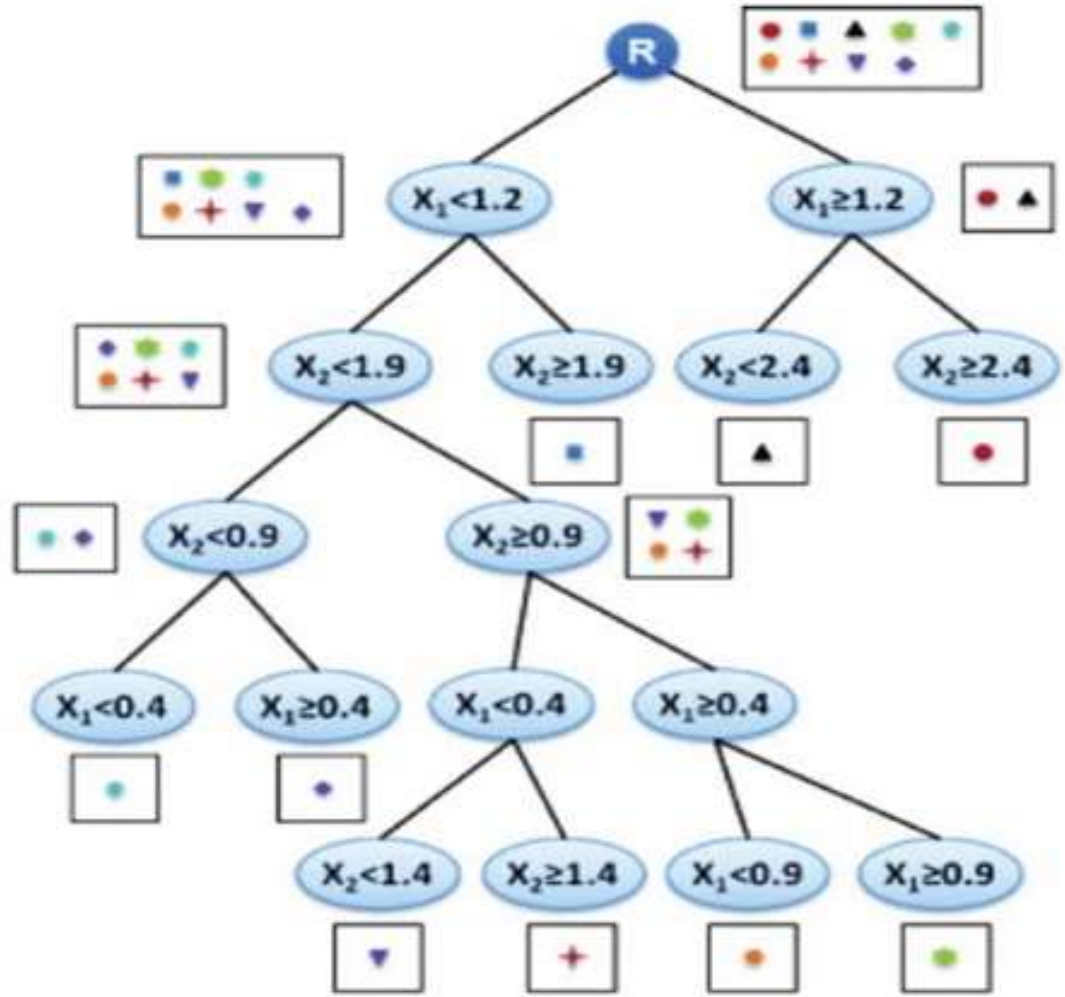







4.0
3.0
2.0
1.0
0.0
0.5
1.0
1.5
2.0
X2
X1
d1
d2
d3
d4
d5
d6
d7
d8
d9

(a)


R
X1<1.2
X1≥1.2
X2<1.9
X2≥1.9
X2<2.4
X2≥2.4
X2<0.9
X2≥0.9
X1<0.4
X1≥0.4
X1<0.4
X1≥0.4
X2<1.4
X2≥1.4
X1<0.9
X1≥0.9

(b)

Adım 1: Hedefin entropisini hesaplanır.

Entropy(PlayGolf) = Entropy (5,9)

= Entropy (0.36, 0.64)

= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)

= 0.94

Adım 2: Veri kümesi daha sonra farklı niteliklere bölünür. Her dal için entropi hesaplanır. Daha sonra, bölme için toplam entropi elde etmek için orantılı olarak eklenir. Ortaya çıkan entropi, bölünmeden önceki entropiden çıkarılır. Sonuç, Bilgi Kazanımı veya entropideki azalmadır.

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

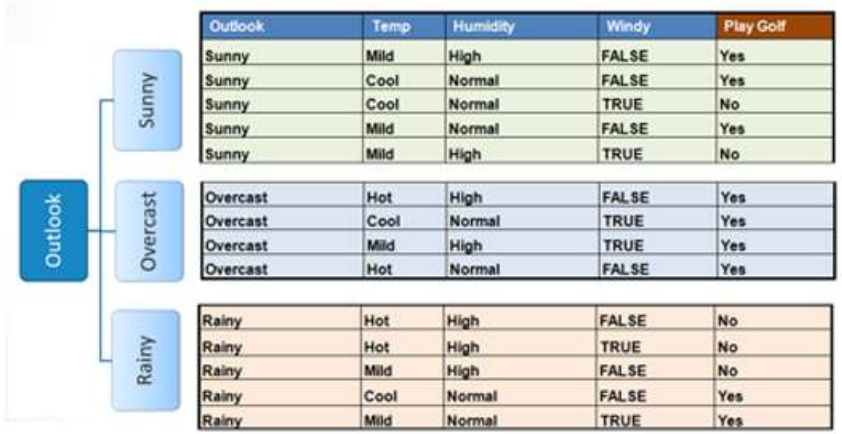
$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$

G(PlayGolf, Outlook) = **E**(PlayGolf) – **E**(PlayGolf, Outlook)

= 0.940 – 0.693 = 0.247

Adım 3: Karar düğümü olarak en büyük bilgi kazancına sahip öznitelik seçilir, veri seti dallarına bölünür ve aynı işlemi her dalda tekrarlanır.

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			



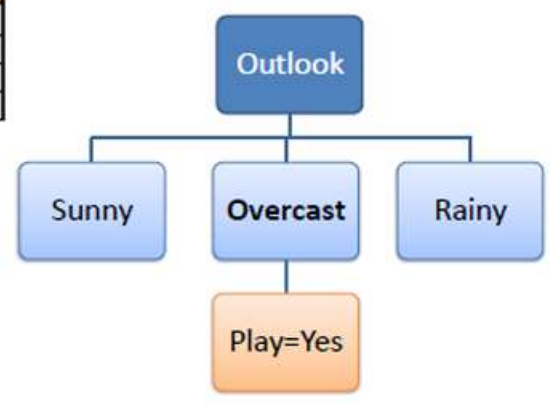
Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No

Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes

Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

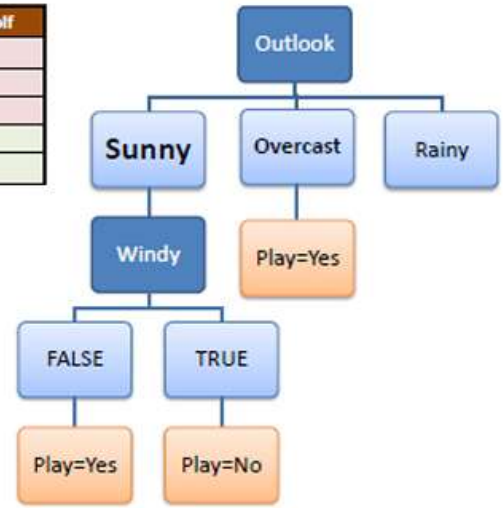
Step 4a: Entropisi 0 olan bir dal, bir yaprak düğümdür.

Temp	Humidity	Windy	Play Golf
Hot	High	FALSE	Yes
Cool	Normal	TRUE	Yes
Mild	High	TRUE	Yes
Hot	Normal	FALSE	Yes



Adım 4b: Entropisi 0'dan büyük olan bir dalın daha fazla bölünmesi gerekir.

Temp	Humidity	Windy	Play Golf
Mild	High	FALSE	Yes
Cool	Normal	FALSE	Yes
Mild	Normal	FALSE	Yes
Cool	Normal	TRUE	No
Mild	High	TRUE	No



Adım 5: ID3 algoritması, tüm veriler sınıflandırılana kadar yaprak olmayan dallarda özyinelemeli olarak çalıştırılır.

Karar Ağacından Karar Kurallarına

Bir karar ağacı, kök düğümden yaprak düğümlere tek tek eşlenerek kolayca bir dizi kurala dönüştürülebilir.

R_1: IF (Outlook=Sunny) AND (Windy=FALSE) THEN Play=Yes

R_2: IF (Outlook=Sunny) AND (Windy=TRUE) THEN Play=No

R_3: IF (Outlook=Overcast) THEN Play=Yes

R_4: IF (Outlook=Rainy) AND (Humidity=High) THEN Play=No

R_5: IF (Outlook=Rain) AND (Humidity=Normal) THEN Play=Yes

File Edit Selection View Go Run Terminal Help

phyton

EXPLORER

PHYTON

- hiyerarşik kümeleme.py
- k-means.py
- karar ağaçları.py
- KNN.py
- phyton denemeleri.py
- pjıopjpo.py

phyton denemeleri.py | pjıopjpo.py | hiyerarşik kümeleme.py | KNN.py | k-means.py | karar ağaçları.py

karar ağaçları.py > ...

```
13  # Veri setini eğitim ve test olarak ayırma
14  X_train, X_test, y_train, y_test = train_test_
15
16  # Karar ağacı sınıflandırıcıyı oluşturma ve eğ
17  clf = DecisionTreeClassifier(random_state=42)
18  clf.fit(X_train, y_train)
19
20  # Test verisi ile tahmin yapma
21  y_pred = clf.predict(X_test)
22
23  # Sonuçları değerlendirme
24  print(f'Accuracy: {accuracy_score(y_test, y_pr
25  print('Confusion Matrix:')
26  print(confusion_matrix(y_test, y_pred))
27  print('Classification Report:')
28  print(classification_report(y_test, y_pred))
29
30  # Karar ağacını görselleştirme
31  plt.figure(figsize=(20, 10))
32  plot_tree(clf, feature_names=iris.feature_name
33  plt.title('Karar Ağacı Görselleştirmesi')
34  plt.show()
35
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      1.00      1.00         9
           2       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

Figure 1