

Data Mining

Classification: Basic Concepts and Techniques, Decision Trees

Lecture Notes for Chapter 4

Introduction to Data Mining, 2nd Edition

by

Tan, Steinbach, Karpatne, Kumar



Classification: Definition

- Bir kayıt koleksiyonu verildiğinde (*training set*)
 - Her kayıt bir çok-öğeli bir veri grubu (x,y) ile karakterize edilir, burada x öznitelik kümesidir ve y sınıf etiketidir.
 - ◆ x : öznitelik (*attribute*), öngösterge (*predictor*), bağımsız değişken (*independent variable*), input
 - ◆ y : sınıf (*class*), response, bağımlı değişken (*dependent variable*), output
- Görev:
 - Her bir öznitelik kümesi x 'i önceden tanımlanmış sınıf etiketlerinden (y) birine eşleyen bir model öğrenmek

Classification: Descriptive Modeling

Bir sınıflandırma modeli, farklı sınıflardan nesneler arasında ayırım yapmak için **açıklayıcı bir araç** görevi görebilir.

Table 4.1. The vertebrate data set. (*Omurgalılar veri seti*)

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hibernates	Class Label
human	warm-blooded	hair	yes	no	no	yes	no	mammal
python	cold-blooded	scales	no	no	no	no	yes	reptile
salmon	cold-blooded	scales	no	yes	no	no	no	fish
whale	warm-blooded	hair	yes	yes	no	no	no	mammal
frog	cold-blooded	none	no	semi	no	yes	yes	amphibian
komodo dragon	cold-blooded	scales	no	no	no	yes	no	reptile
bat	warm-blooded	hair	yes	no	yes	yes	yes	mammal
pigeon	warm-blooded	feathers	no	no	yes	yes	no	bird
cat	warm-blooded	fur	yes	no	no	yes	no	mammal
leopard	cold-blooded	scales	yes	yes	no	no	no	fish
shark								
turtle	cold-blooded	scales	no	semi	no	yes	no	reptile
penguin	warm-blooded	feathers	no	semi	no	yes	no	bird
porcupine	warm-blooded	quills	yes	no	no	yes	yes	mammal
eel	cold-blooded	scales	no	yes	no	no	no	fish
salamander	cold-blooded	none	no	semi	no	yes	yes	amphibian

Classification: Predictive Modeling

Predictive Modeling (Öngörücü modelleme) Bir sınıflandırma modeli, bilinmeyen kayıtların sınıf etiketini tahmin etmek için de kullanılabilir. Şekil 4.2'de gösterildiği gibi, bir sınıflandırma modeli, bilinmeyen bir kaydın öznitelik kümesiyle sunulduğunda otomatik olarak bir sınıf etiketi atayan bir kara kutu (**black box**) olarak değerlendirilebilir.

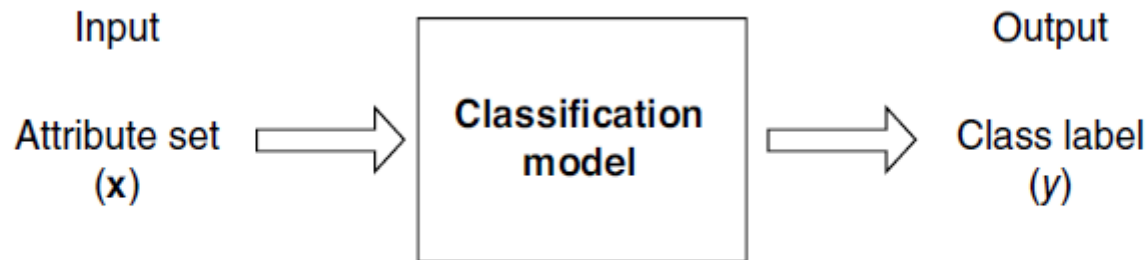


Figure 4.2. Classification as the task of mapping an input attribute set x into its class label y .

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hibernates	Class Label
gila monster	cold-blooded	scales	no	no	no	yes	yes	?

Examples of Classification Task

Task	Attribute set, x	Class label, y
Categorizing email messages	Features extracted from email message header and content	spam or non-spam
Identifying tumor cells	Features extracted from MRI scans	malignant or benign cells
Cataloging galaxies	Features extracted from telescope images	Elliptical, spiral, or irregular-shaped galaxies

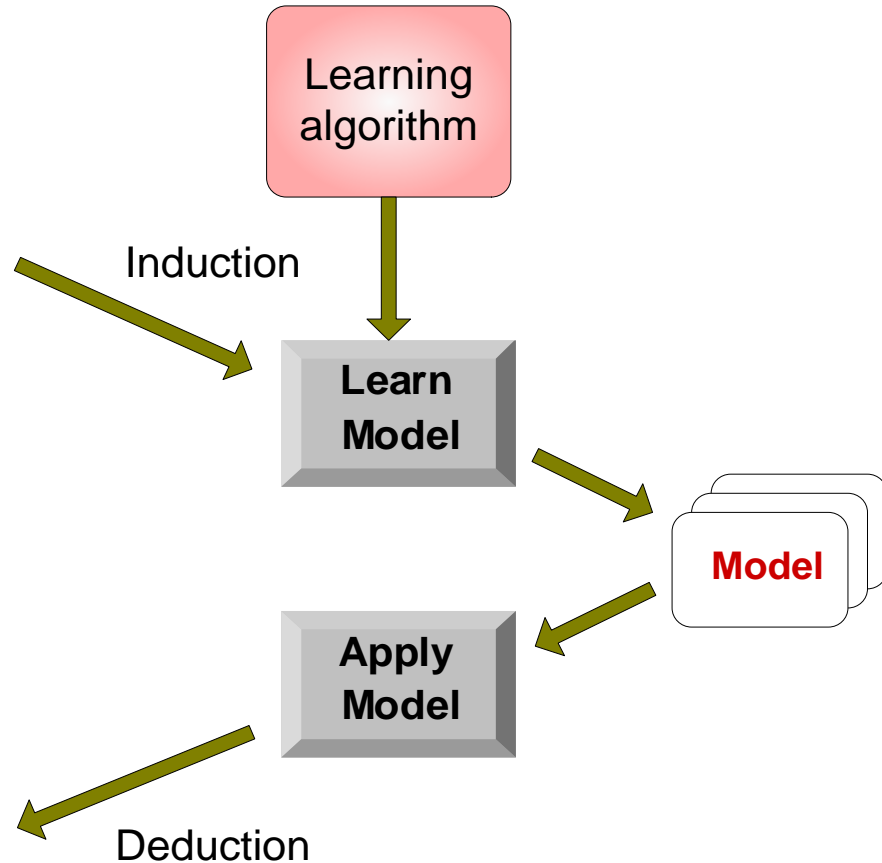
General Approach for Building Classification Model

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Classification Techniques

- Base Classifiers
 - Decision Tree based Methods
 - Rule-based Methods
 - Nearest-neighbor
 - Neural Networks
 - Deep Learning
 - Naïve Bayes and Bayesian Belief Networks
 - Support Vector Machines
- Ensemble Classifiers
 - Boosting, Bagging, Random Forests

Decision Tree Induction: How a Decision Tree Works

- Ağacın üç tür düğümü vardır:
 - Giren kenarı olmayan (**no incoming edges**) ve sıfır veya daha fazla çıkan kenarı olan kök düğüm (**root node**)
 - İç düğüm (**Internal nodes**), Her biri tam olarak bir giren kenara (**one incoming edge**) ve iki veya daha fazla çıkan kenara (**two or more outgoing edges**) sahip olan iç düğümler.
 - Yaprak veya uç düğümler (**Leaf or terminal nodes**), her biri tam olarak bir giren kenara (**one incoming edge**) sahiptir ve çıkan kenarı yoktur (**no outgoing edges**).

Decision Tree Induction: How a Decision Tree Works

- Bir karar ağacında, her **yaprak düğüme** bir **sınıf etiketi** atanır.
- Kök ve diğer iç düğümleri içeren uç-birim olmayan düğümler (**nonterminal nodes**), farklı özelliklere sahip kayıtları ayırmak için öznitelik test koşullarını (**attribute test conditions**) içerir.
 - Örneğin, Şekil 4.4'te gösterilen kök düğüm, sıcakkanlıları (**warm-blooded**) soğukkanlı (**cold-blooded**) omurgalılardan ayırmak için Body Temperature öz niteliğini kullanır.

Decision Tree Induction: How a Decision Tree Works

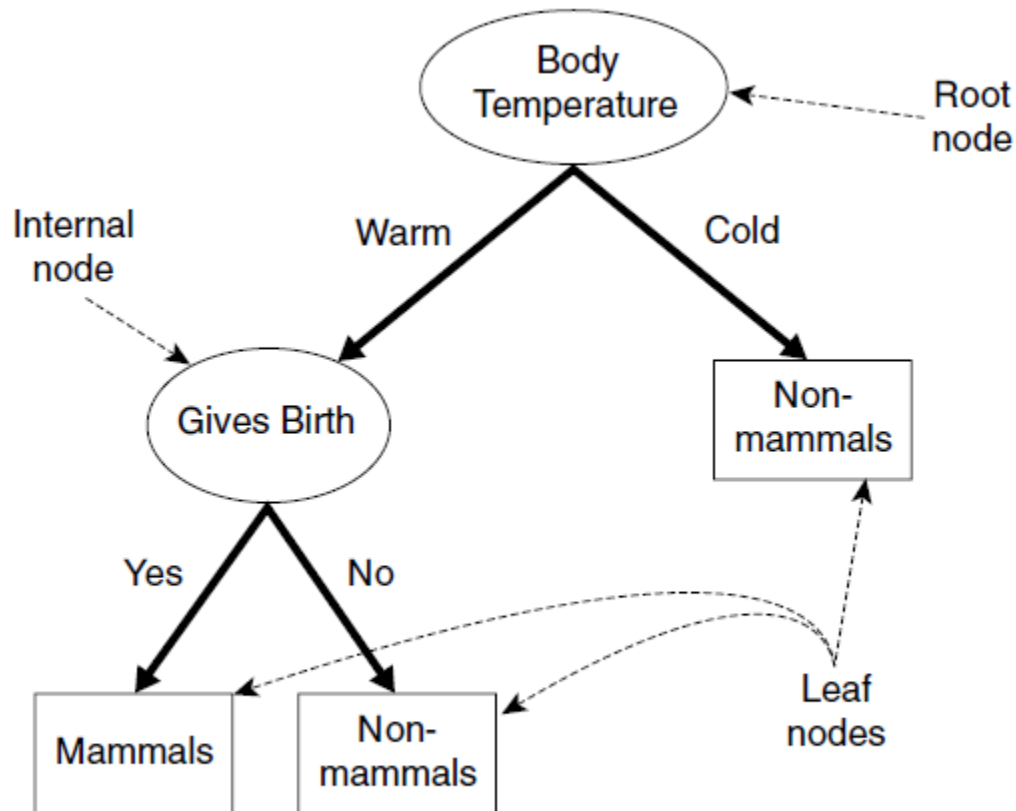


Figure 4.4. A decision tree for the mammal classification problem.

Tüm soğukkanlı omurgalılar **non-mammals** olduğu için, kök düğümün sağ çocuğu olarak Non-mammals etiketli bir yaprak düğümü oluşturulur.

Bir karar ağacı oluşturulduktan sonra, **bir test kaydının sınıflandırılması** kolaydır. Kök düğümünden başlayarak, **test koşulunu kayda uygularız** ve testin sonucuna göre uygun dalı takip ederiz.

Decision Tree Induction: How a Decision Tree Works

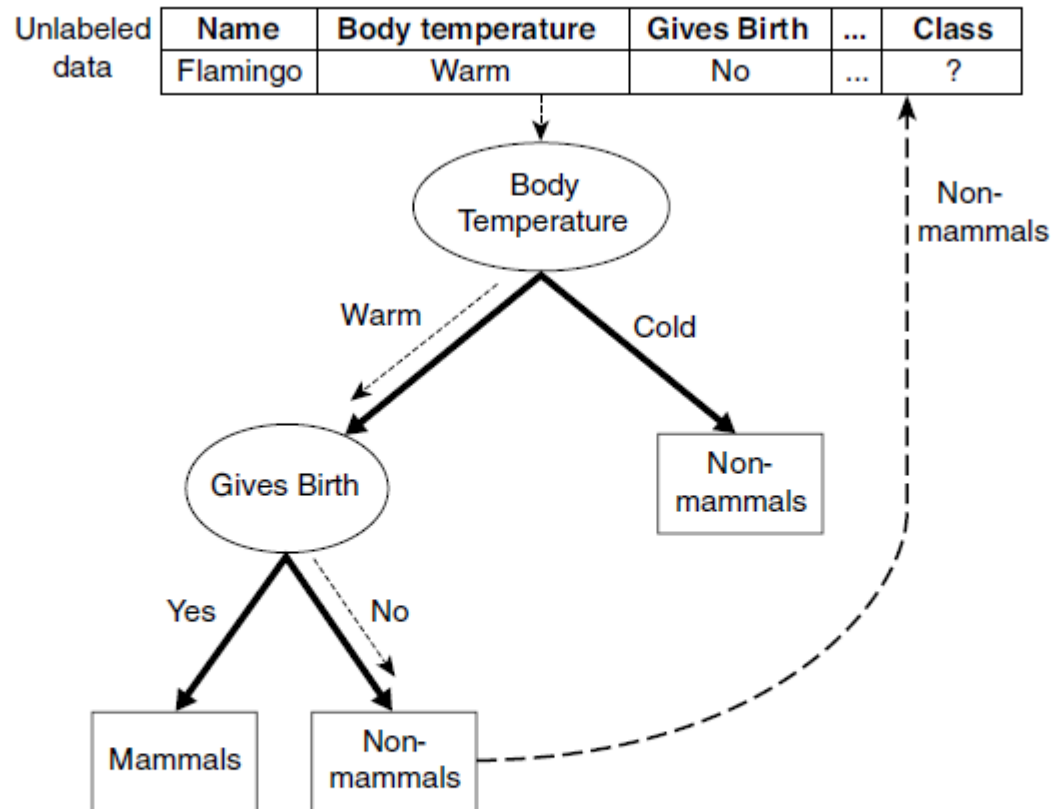


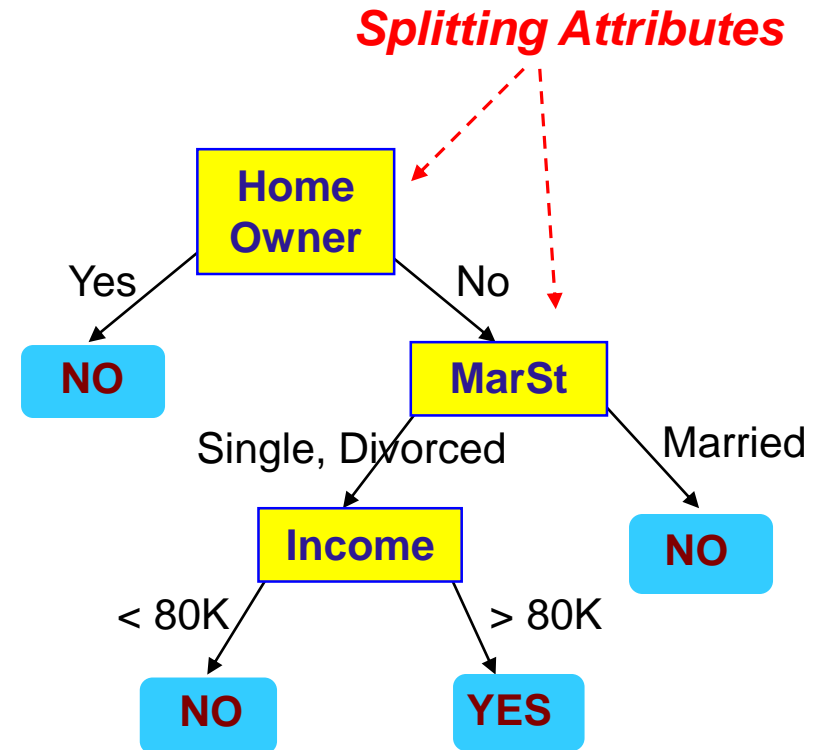
Figure 4.5. Classifying an unlabeled vertebrate. The dashed lines represent the outcomes of applying various attribute test conditions on the unlabeled vertebrate. The vertebrate is eventually assigned to the `Non-mammal` class.

Example of a Decision Tree

categorical
categorical
continuous
class

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data

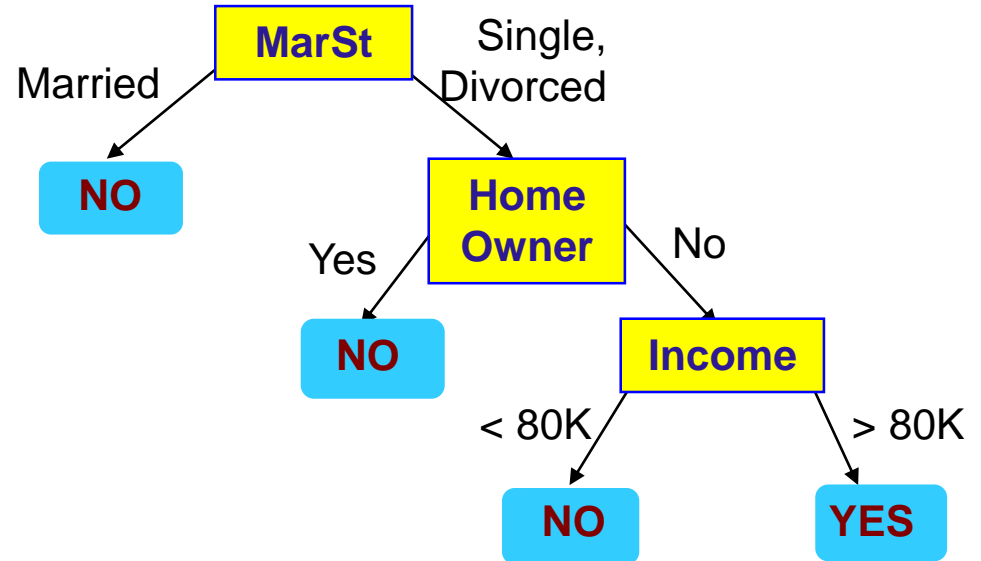


Model: Decision Tree

Another Example of Decision Tree

categorical
categorical
continuous
class

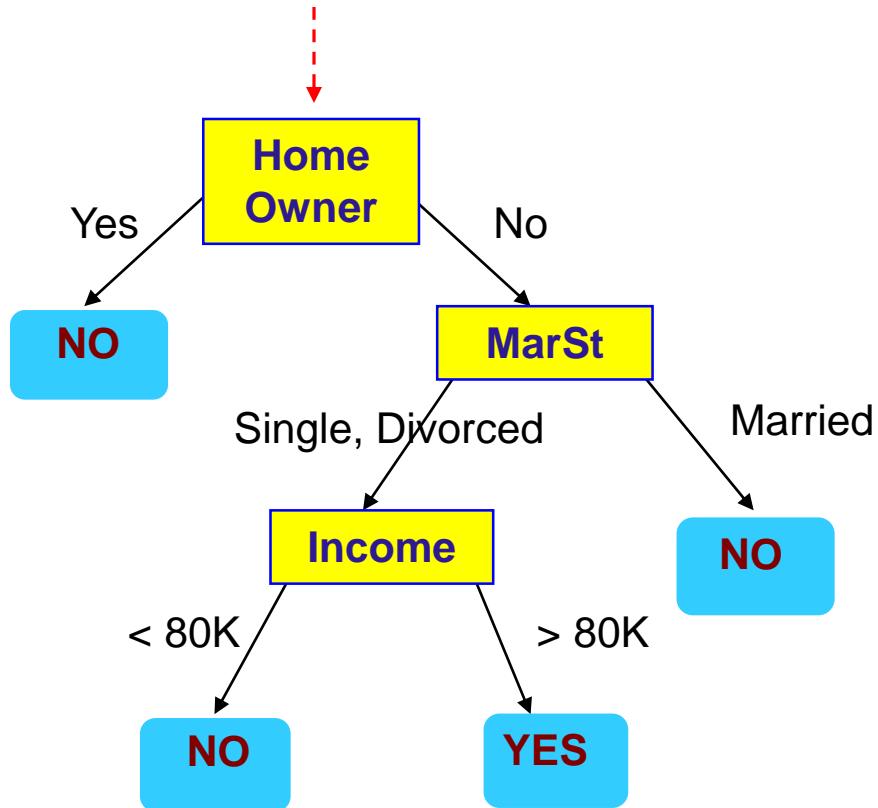
ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Aynı verilere uyan birden fazla ağaç olabilir!

Apply Model to Test Data

Start from the root of tree.



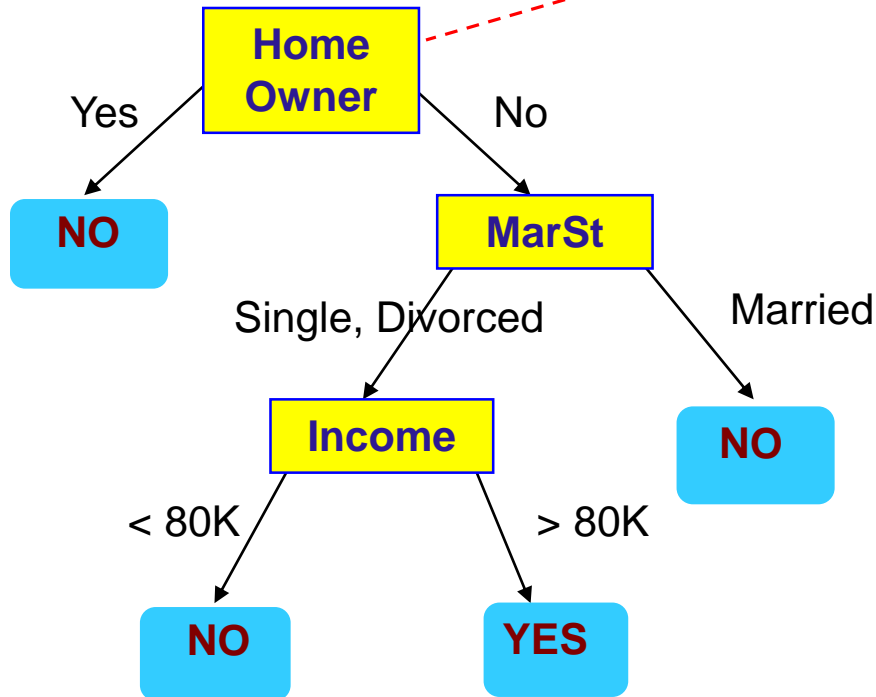
Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

Apply Model to Test Data

Test Data

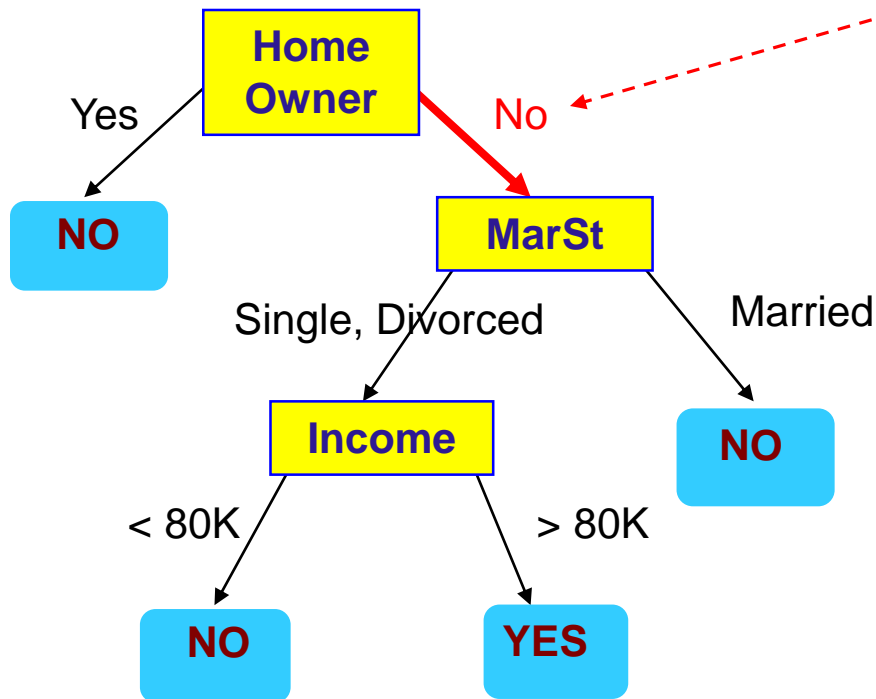
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Apply Model to Test Data

Test Data

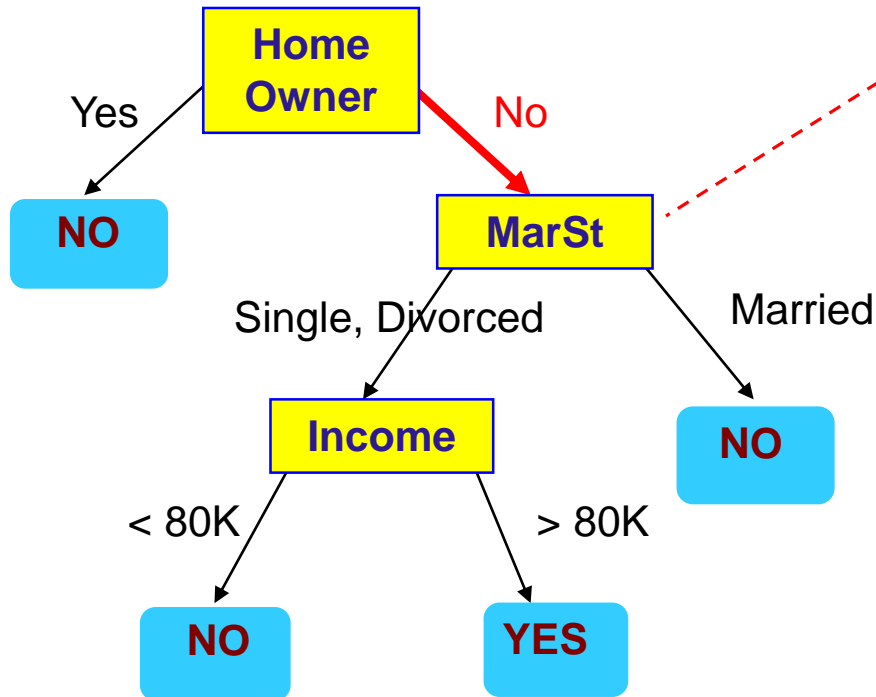
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Apply Model to Test Data

Test Data

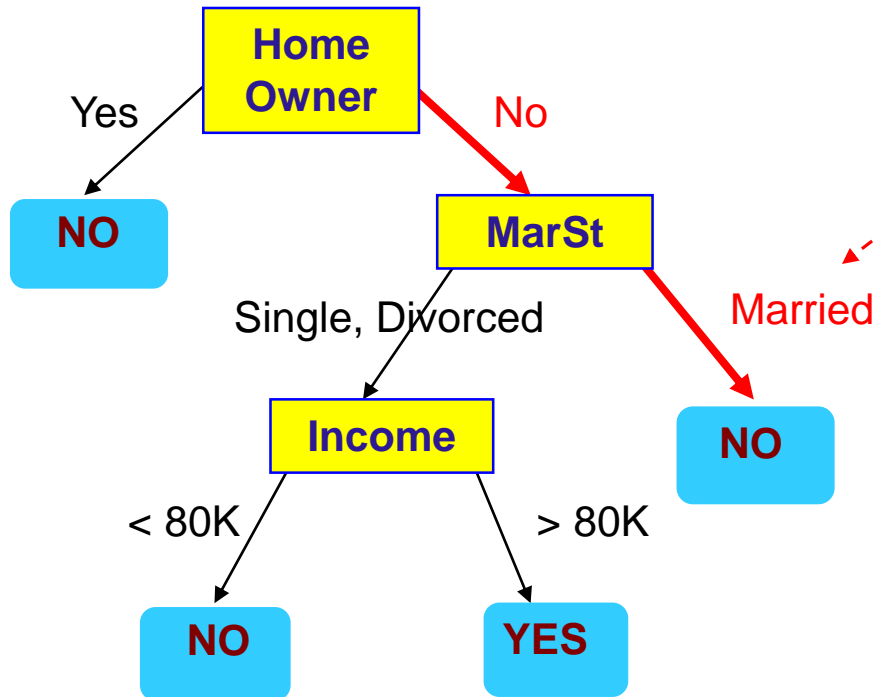
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Apply Model to Test Data

Test Data

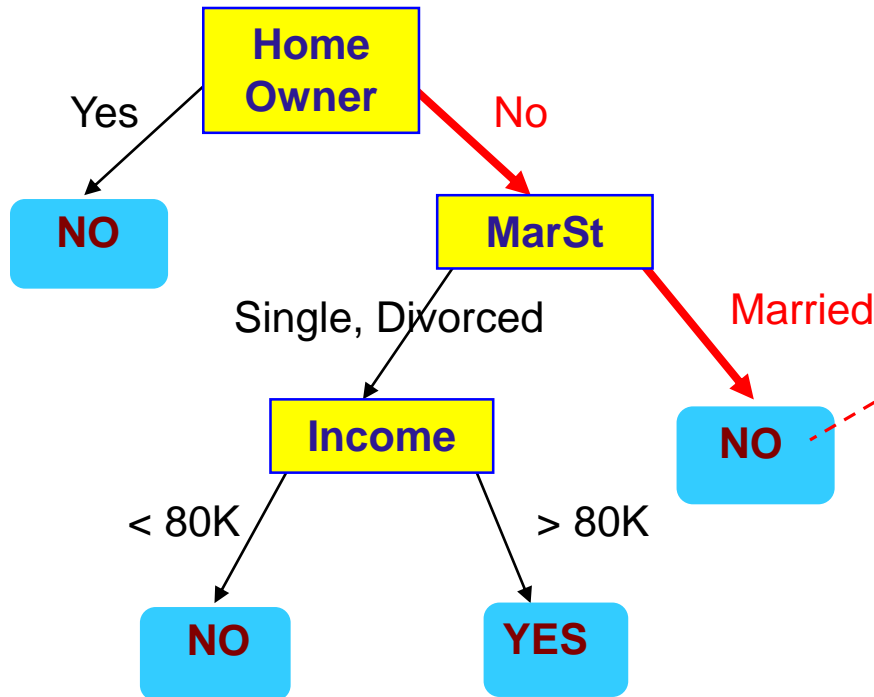
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Apply Model to Test Data

Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Assign Defaulted to
"No"

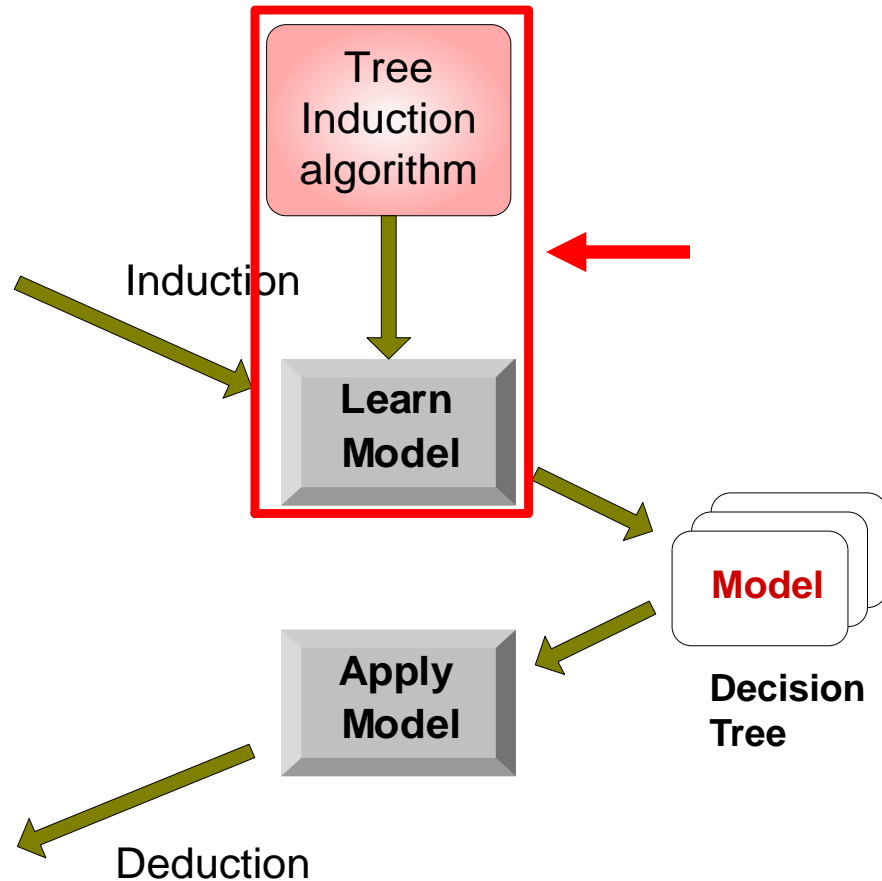
Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



How to Build a Decision Tree

- Prensipte, belirli bir özellik kümesinden oluşturulabilen **üstel olarak çok sayıda karar ağacı** vardır.
- Ağaçların bazıları diğerlerinden daha doğru olsa da, **en uygun ağacı bulmak, arama uzayının üstel boyutu nedeniyle hesaplama açısından mümkün değildir.**
- Bununla birlikte, **verimli algoritmalar makul bir süre içinde** (optimal olmasa da) makul ölçüde doğru bir karar ağacı oluşturmak için geliştirilmiştir.
- Bu algoritmalar genellikle **verileri bölmek için hangi öz niteliğin** kullanılacağına dair bir dizi yerel olarak optimum kararlar alarak bir karar ağacı oluşturan açgözlü bir strateji (**greedy strategy**) kullanır.

Decision Tree Induction

- Many Algorithms:
 - Hunt's Algorithm (one of the earliest)
 - CART
 - ID3, C4.5
 - SLIQ, SPRINT

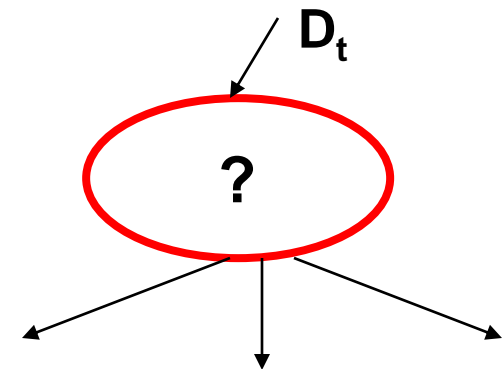
General Structure of Hunt's Algorithm

- Hunt'ın algoritmasında, eğitim kayıtlarını ardışık olarak daha saf alt kümelere (**purier subsets**) bölerek yinelemeli bir şekilde (**in a recursive fashion**) bir karar ağacı büyütülür.
 - D_t , t düğümü ile ilişkili eğitim kayıtları kümesi olsun ve
 - $y = \{y_1, y_2, \dots, y_C\}$ sınıf etiketleri olsun. Aşağıda, Hunt algoritmasının yinelemeli bir tanımı bulunmaktadır.

General Structure of Hunt's Algorithm

- | D_t , t düğümüne ulaşan eğitim kayıtları kümesi olsun
- | Genel Prosedür:
 - Eğer D_t , **aynı sınıfa** (y_t) ait kayıtları içeriyorsa t , y_t olarak etiketlenmiş bir yaprak düğümdür (**leaf node**).
 - Eğer D_t **birden fazla sınıfa** ait kayıtlar içeriyorsa, verileri daha **küçük alt kümelere** (oluşturulan alt düğümler- **child nodes**) bölmek için bir öznitelik testi kullanın. **Prosedürü her alt kümeye yinelemeli olarak uygulayın.**

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Hunt's Algorithm

Defaulted = No

(7,3)

(a)

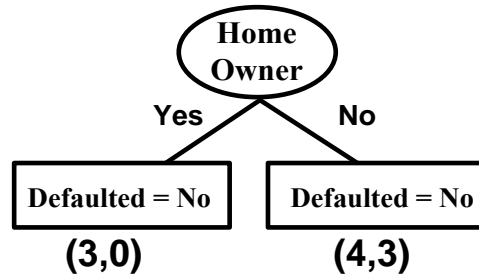
ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Hunt's Algorithm

Defaulted = No

(7,3)

(a)



(b)

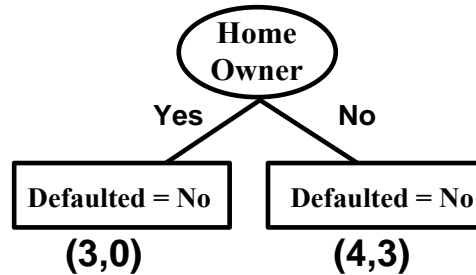
ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Hunt's Algorithm

Defaulted = No

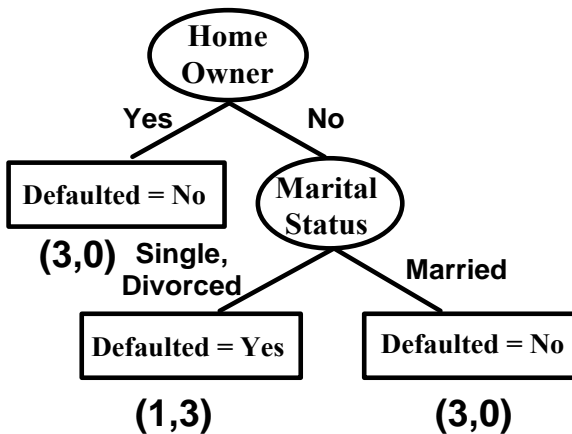
(7,3)

(a)



(b)

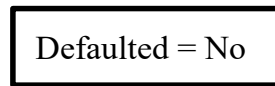
ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



(c)

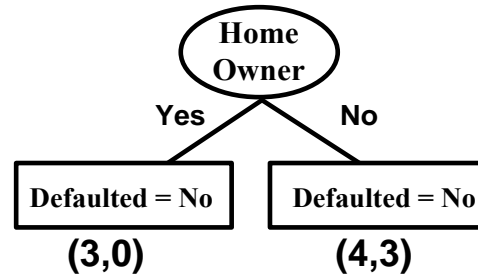
Hunt's Algorithm

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

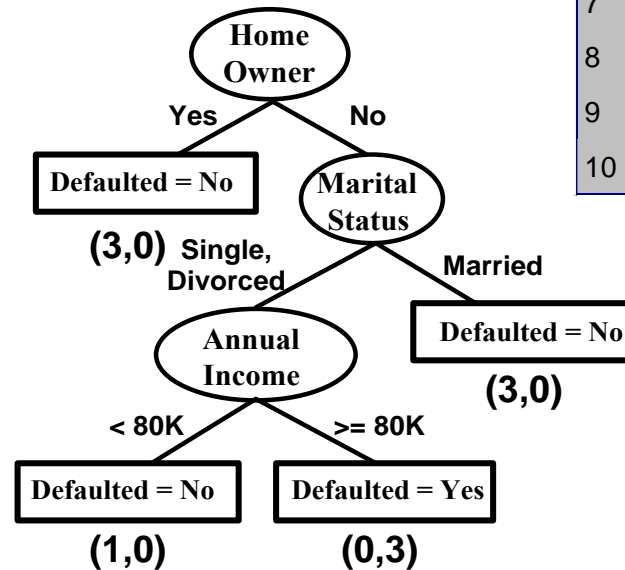


(7,3)

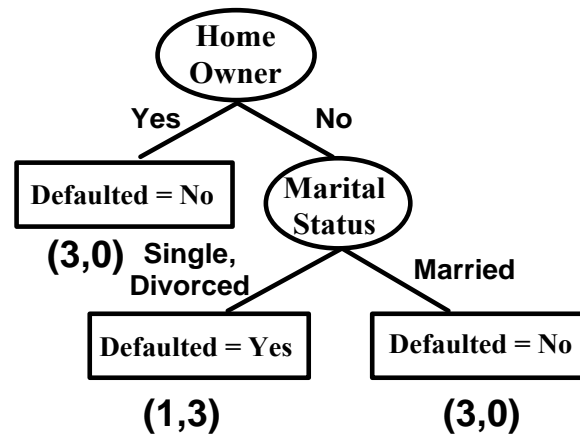
(a)



(b)



(d)



(c)

Design Issues of Decision Tree Induction

- | Eğitim dataları/kayıtları nasıl bölünmeli?
 - ◆ öz nitelik türlerine bağlı olarak
 - Bir test koşulunun iyiliğini (*goodness*) değerlendirmek için ölçüt
- | Bölme prosedürü (*splitting procedure*) nasıl durdurulmalı?
 - Tüm kayıtlar aynı sınıfa aitse veya aynı öz nitelik değerlerine sahipse bölmeyi durdurun
 - Erken sonlandırma (*early termination*)

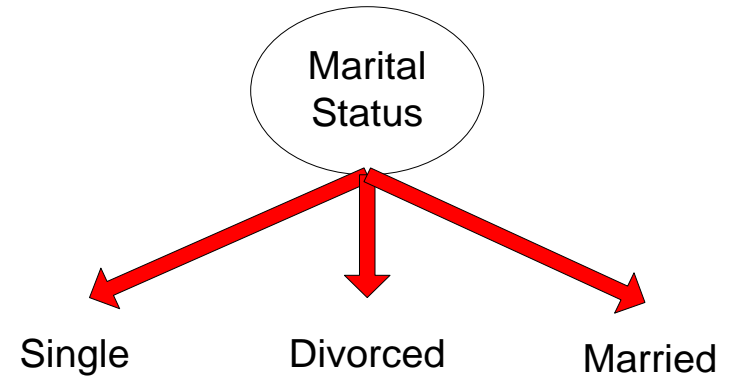
Methods for Expressing Test Conditions

- | Depends on attribute types
 - Binary
 - Nominal
 - Ordinal
 - Continuous
- | Depends on number of ways to split
 - 2-way split
 - Multi-way split

Test Condition for Nominal Attributes

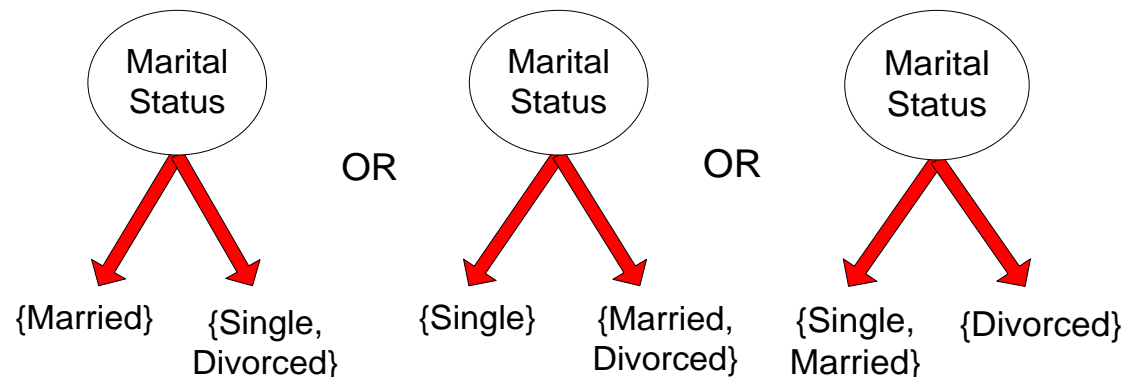
- Multi-way split:

- Farklı değerlerin (*distinct values*) sayısı kadar bölüm kullanır



- Binary split:

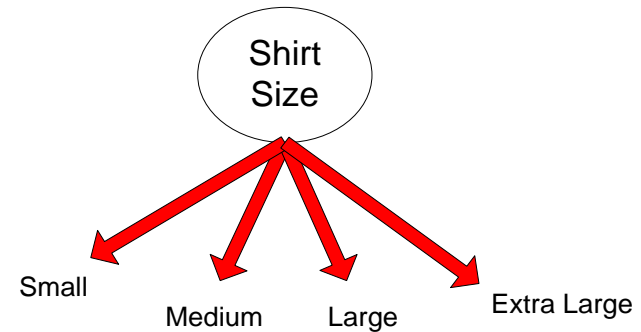
- Değerleri iki alt gruba ayırır



Test Condition for Ordinal Attributes

| Multi-way split:

- Farklı değerlerin (*distinct values*) sayısı kadar bölüm kullanır

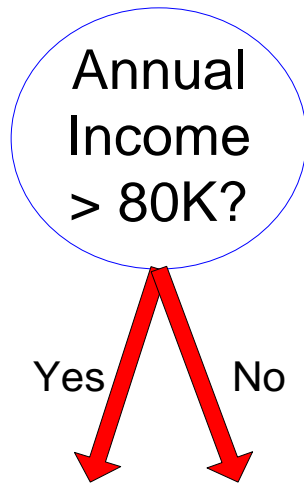


| Binary split:

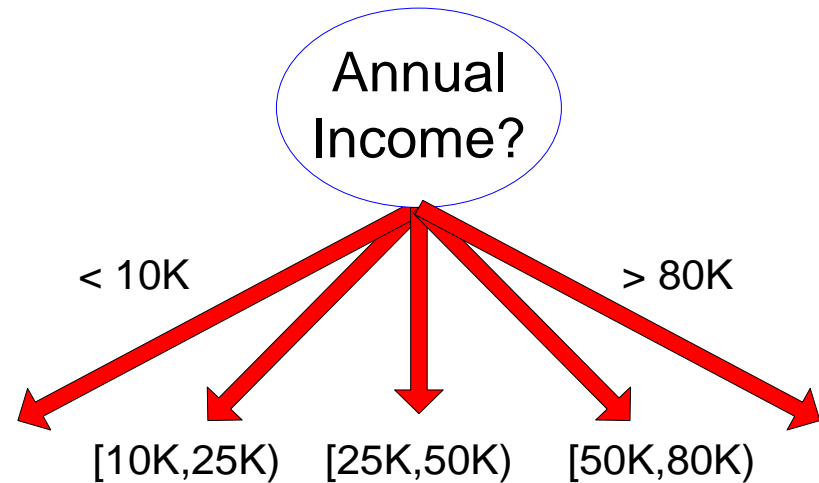
- Değerleri iki alt gruba ayırır
- Öznitelik değerleri arasında sıra özelliğini (*order property*) korunur



Test Condition for Continuous Attributes



(i) Binary split



(ii) Multi-way split

Splitting Based on Continuous Attributes

- Farklı şekillerde ele alınabilir
 - Sıralı bir kategorik öznelik (*ordinal categorical attribute*) oluşturmak için ayrıklaştırma (**Discretization**)

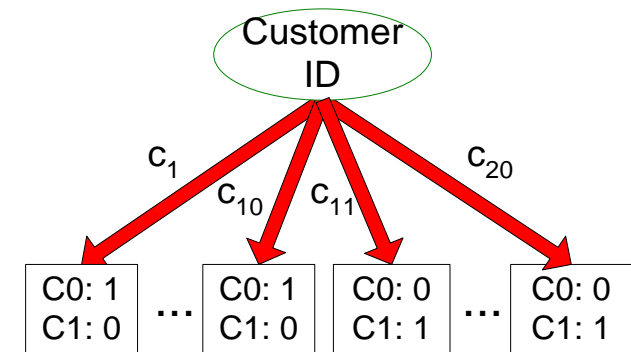
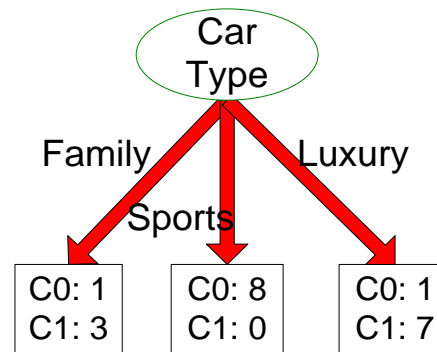
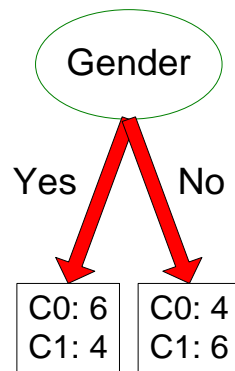
Aralıklar (*ranges*), eşit aralıklı kümeleme, eşit sıklıkta kümeleme (yüzdelikler) veya kümeleme ile bulunabilir.

 - ◆ Static – discretize once at the beginning
 - ◆ Dynamic – repeat at each node
 - **Binary Decision**: $(A < v)$ or $(A \geq v)$
 - ◆ olası tüm bölümlmeleri düşünüp en iyi kesimi bulur
 - ◆ daha yoğun işlem gerektirebilir

How to determine the Best Split

**Before Splitting: 10 records of class 0,
10 records of class 1**

Customer Id	Gender	Car Type	Shirt Size	Class
1	M	Family	Small	C0
2	M	Sports	Medium	C0
3	M	Sports	Medium	C0
4	M	Sports	Large	C0
5	M	Sports	Extra Large	C0
6	M	Sports	Extra Large	C0
7	F	Sports	Small	C0
8	F	Sports	Small	C0
9	F	Sports	Medium	C0
10	F	Luxury	Large	C0
11	M	Family	Large	C1
12	M	Family	Extra Large	C1
13	M	Family	Medium	C1
14	M	Luxury	Extra Large	C1
15	F	Luxury	Small	C1
16	F	Luxury	Small	C1
17	F	Luxury	Medium	C1
18	F	Luxury	Medium	C1
19	F	Luxury	Medium	C1
20	F	Luxury	Large	C1



Which test condition is the best?

How to determine the Best Split

- | Greedy approach:
 - Nodes with **pur**er class distribution are preferred
- | Need a measure of node impurity:

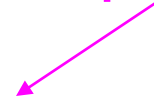
C0: 5
C1: 5

High degree of impurity

C0: 9
C1: 1

Low degree of impurity

pur



Measures of Node Impurity

| Gini Index

$$GINI(t) = 1 - \sum_j [p(j | t)]^2$$

| Entropy

$$Entropy(t) = -\sum_j p(j | t) \log p(j | t)$$

| Misclassification error

$$Error(t) = 1 - \max_i P(i | t)$$

Finding the Best Split

1. Compute impurity measure (P) before splitting
2. Compute impurity measure (M) after splitting
 - | Compute impurity measure of each child node
 - | M is the weighted impurity of children
3. Choose the attribute test condition that produces the **highest gain**

$$\text{Gain} = P - M$$

or equivalently, **lowest impurity** measure after splitting (M)

Finding the Best Split

Before Splitting:

C0	N00
C1	N01

→ P

A?

Yes

No

Node N1

Node N2

C0 **N10**

C1 **N11**

C0 **N20**

C1 **N21**

↓
M11

↓
M12

M1

B?

Yes

No

Node N3

Node N4

C0 **N30**

C1 **N31**

C0 **N40**

C1 **N41**

↓
M21

↓
M22

M2

Gain = P – M1 vs P – M2

Measure of Impurity: GINI

- Gini Index for a given node t :

$$GINI(t) = 1 - \sum_j [p(j | t)]^2$$

(NOTE: $p(j | t)$ is the relative frequency of class j at node t).

- **Maximum** ($1 - 1/n_c$) when records are **equally distributed** among all classes, implying **least interesting information**
- **Minimum** (0.0) when all records belong to **one class**, implying **most interesting information**

Measure of Impurity: GINI

- Gini Index for a given node t :

$$GINI(t) = 1 - \sum_j [p(j | t)]^2$$

(NOTE: $p(j | t)$ is the relative frequency of class j at node t).

- For 2-class problem ($p, 1 - p$):

- ◆ $GINI = 1 - p^2 - (1 - p)^2 = 2p(1-p)$

C1	0
C2	6
Gini=0.000	

C1	1
C2	5
Gini=0.278	

C1	2
C2	4
Gini=0.444	

C1	3
C2	3
Gini=0.500	

Computing Gini Index of a Single Node

$$GINI(t) = 1 - \sum_j [p(j | t)]^2$$

C1	0
C2	6

$$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$$

$$Gini = 1 - P(C1)^2 - P(C2)^2 = 1 - 0 - 1 = 0$$

C1	1
C2	5

$$P(C1) = 1/6 \quad P(C2) = 5/6$$

$$Gini = 1 - (1/6)^2 - (5/6)^2 = 0.278$$

C1	2
C2	4

$$P(C1) = 2/6 \quad P(C2) = 4/6$$

$$Gini = 1 - (2/6)^2 - (4/6)^2 = 0.444$$

Computing Gini Index for a Collection of Nodes

- | When a node p is split into k partitions (children)

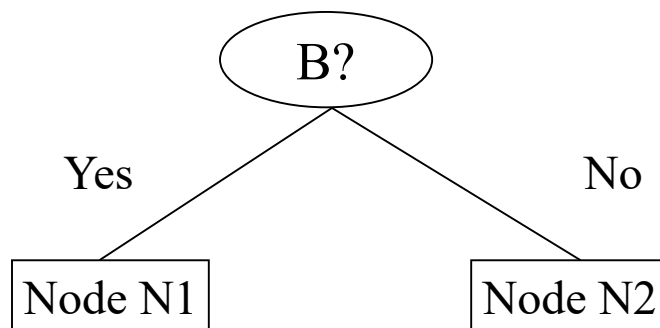
$$GINI_{split} = \sum_{i=1}^k \frac{n_i}{n} GINI(i)$$

where, n_i = number of records at child i ,
 n = number of records at parent node p .

- | Choose the attribute that **minimizes weighted average Gini index of the children**
- | Gini index is used in decision tree algorithms such as CART, SLIQ, SPRINT

Binary Attributes: Computing GINI Index

- Splits into two partitions
- Effect of Weighing partitions:
 - **Larger and Purer Partitions** are sought for.



	Parent
C1	7
C2	5
Gini = 0.486	

Gini(N1)

$$= 1 - (5/6)^2 - (1/6)^2$$
$$= 0.278$$

Gini(N2)

$$= 1 - (2/6)^2 - (4/6)^2$$
$$= 0.444$$

	N1	N2
C1	5	2
C2	1	4
Gini=0.361		

Weighted Gini of N1 N2

$$= 6/12 * 0.278 +$$
$$6/12 * 0.444$$
$$= 0.361$$

$$\text{Gain} = 0.486 - 0.361 = 0.125$$

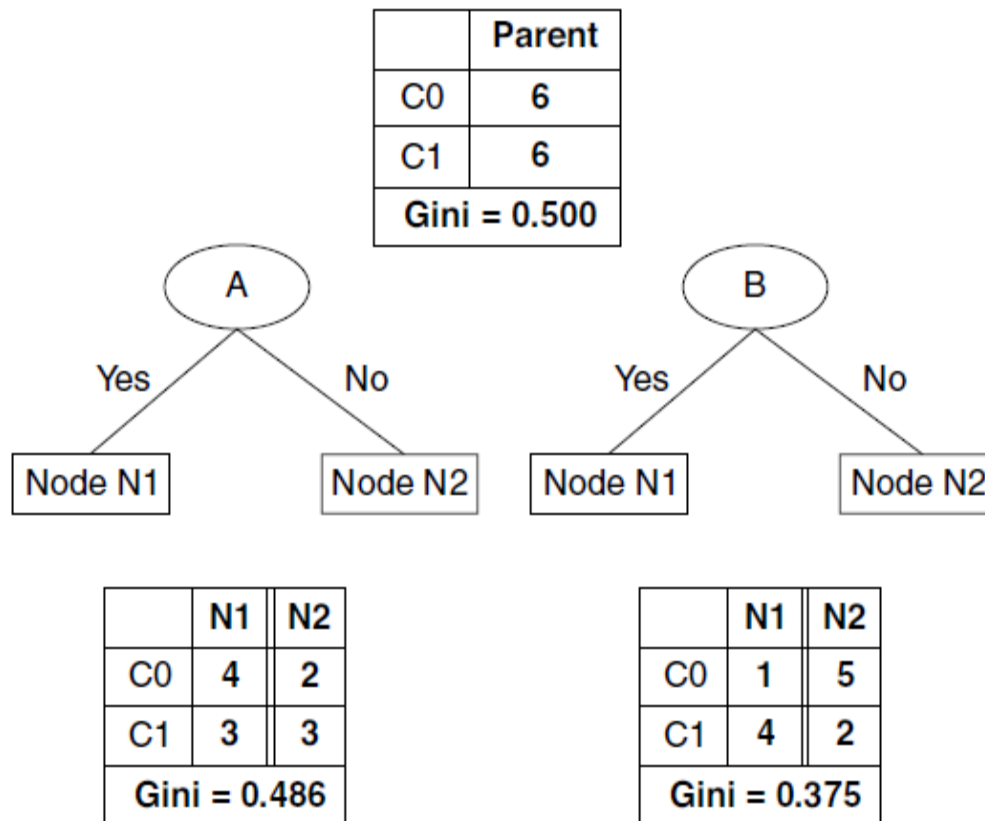


Figure 4.14. Splitting binary attributes.

- Bölünmeden önce, her iki sınıftan eşit sayıda kayıt olduğundan Gini indeksi 0.5'tir.
- Verileri bölmek için A özniteliği seçilirse, N1 düğümü için Gini indeksi 0,490 ve N2 düğümü için 0,480'dir.
- Alt düğümler için Gini indeksinin ağırlıklı ortalaması $(7/12) \times 0.4898 + (5/12) \times 0.480 = 0.486$.
- Benzer şekilde, B özniteliği için Gini indeksinin ağırlıklı ortalamasının 0,375 olduğunu gösterebiliriz.
- B özniteliğinin alt kümeleri (*subsets*) **daha küçük bir Gini indeksine sahip olduğundan, A özniteliği yerine tercih edilir.**

Categorical Attributes: Computing Gini Index

- Her farklı değer için, veri kümesindeki her bir sınıfın sayılarını toplayın
- Karar vermek için sayım matrisini (*count matrix*) kullanın

Multi-way split

	CarType		
	Family	Sports	Luxury
C1	1	8	1
C2	3	0	7
Gini	0.163		

Two-way split
(find best partition of values)

	CarType	
	{Sports, Luxury}	{Family}
C1	9	1
C2	7	3
Gini	0.468	

	CarType	
	{Sports}	{Family, Luxury}
C1	8	2
C2	0	10
Gini	0.167	

«Multiway split», her iki «Two-way split» kıyasla daha küçük bir Gini indeksine sahiptir.

Which of these is the best?

Üçüncü gruptamanın daha düşük bir Gini indeksi vardır çünkü karşılık gelen alt kümeleri çok daha saftır.

Continuous Attributes: Computing Gini Index

- | Tek bir değere dayalı İkili Kararlar (*Binary Decisions*) kullanın
- | Bölme değeri (*splitting value*) için birçok seçenek
 - Olası bölme değerlerinin sayısı = Farklı değerlerin sayısı
- | Her bölme değerinin kendisiyle ilişkili bir sayım matrisi vardır
 - Her bölümdeki (partitions) sınıf sayıları, $A < v$ ve $A \geq v$
- | En iyi v 'yi seçmek için basit yöntem
 - Her v için, sayım matrisini toplamak ve Gini indeksini hesaplamak için veritabanını tarayın
 - Computationally Inefficient! Repetition of work.

ID	Home Owner	Marital Status	Annual Income	Defaulted
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Annual Income ?

≤ 80 > 80

Defaulted Yes

0	3
---	---

Defaulted No

3	4
---	---



Continuous Attributes: Computing Gini Index...

- Verimli hesaplama için: her öznitelik için,
 - Öznitelik değerlerini sıralayın
 - Her seferinde sayım matrisini güncelleyerek ve gini indeksini hesaplayarak bu değerleri doğrusal olarak tarayın
 - **En düşük gini indeksine** sahip bölme konumu (*split position*) seçin

Sorted Values →	Defaulted	No	No	No	Yes	Yes	Yes	No	No	No	No
	Annual Income										
	60	70	75	85	90	95	100	120	125	220	

Continuous Attributes: Computing Gini Index...

- Verimli hesaplama için: her öznitelik için,
 - Öznitelik değerlerini sıralayın
 - Her seferinde sayım matrisini güncelleyerek ve gini indeksini hesaplayarak bu değerleri doğrusal olarak tarayın
 - **En düşük gini indeksine** sahip bölme konumu (*split position*) seçin

Sorted Values Split Positions	 	Defaulted	<table><tr><td>No</td><td>No</td><td>No</td><td>Yes</td><td>Yes</td><td>Yes</td><td>No</td><td>No</td><td>No</td><td>No</td></tr><tr><td colspan="10">Annual Income</td></tr><tr><td>60</td><td>70</td><td>75</td><td>85</td><td>90</td><td>95</td><td>100</td><td>120</td><td>125</td><td>220</td></tr><tr><td>55</td><td>65</td><td>72</td><td>80</td><td>87</td><td>92</td><td>97</td><td>110</td><td>122</td><td>172</td><td>230</td></tr><tr><td><=</td><td>></td><td><=</td><td>></td><td><=</td><td>></td><td><=</td><td>></td><td><=</td><td>></td><td><=</td><td>></td></tr></table>												No	No	No	Yes	Yes	Yes	No	No	No	No	Annual Income										60	70	75	85	90	95	100	120	125	220	55	65	72	80	87	92	97	110	122	172	230	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
		No	No	No	Yes	Yes	Yes	No	No	No	No																																																								
		Annual Income																																																																	
		60	70	75	85	90	95	100	120	125	220																																																								
		55	65	72	80	87	92	97	110	122	172	230																																																							
<=	>	<=	>	<=	>	<=	>	<=	>	<=	>																																																								

Candidate split positions are identified by taking the **midpoints** between two adjacent sorted values: 55, 65, 72, and so on.

Continuous Attributes: Computing Gini Index...

- Verimli hesaplama için: her öznitelik için,
 - Öznitelik değerlerini sıralayın
 - Her seferinde sayım matrisini güncelleyerek ve gini indeksini hesaplayarak bu değerleri doğrusal olarak tarayın
 - **En düşük gini indeksine** sahip bölme konumu (*split position*) seçin

seçim

Defaulted

No

No

No

Yes

Yes

Yes

No

No

No

No

Sorted Values

→

60

70

75

85

90

95

100

120

125

220

Split Positions

→

55

<=

>

65

<=

>

72

<=

>

80

<=

>

87

<=

>

92

<=

>

97

<=

>

110

<=

>

122

<=

>

172

<=

>

230

<=

>

Yes

No

Gini

0

3

3

4

0.343

Continuous Attributes: Computing Gini Index...

- Verimli hesaplama için: her öznitelik için,
 - Öznitelik değerlerini sıralayın
 - Her seferinde sayım matrisini güncelleyerek ve gini indeksini hesaplayarak bu değerleri doğrusal olarak tarayın
 - **En düşük gini indeksine** sahip bölme konumu (*split position*) seçin

seçim

Defaulted

No

No

No

Yes

Yes

Yes

No

No

No

No

Sorted Values

60

70

75

85

90

95

100

120

125

220

Split Positions

55

<=

>

65

<=

>

72

<=

>

80

<=

>

87

<=

>

92

<=

>

97

<=

>

110

<=

>

122

<=

>

172

<=

>

230

<=

>

Yes

0

3

1

2

No

3

4

3

4

Gini

0.343

0.417

Continuous Attributes: Computing Gini Index...

- Verimli hesaplama için: her öznitelik için,
 - Öznitelik değerlerini sıralayın
 - Her seferinde sayım matrisini güncelleyerek ve gini indeksini hesaplayarak bu değerleri doğrusal olarak tarayın
 - **En düşük gini indeksine** sahip bölme konumu (*split position*) seçin

Sorted Values Split Positions	Defaulted	<table><tr><td colspan="2">No</td><td colspan="2">No</td><td colspan="2">No</td><td colspan="2">Yes</td><td colspan="2">Yes</td><td colspan="2">Yes</td><td colspan="2">No</td><td colspan="2">No</td><td colspan="2">No</td><td colspan="2">No</td></tr><tr><td colspan="20">Annual Income</td></tr><tr><td colspan="2">60</td><td colspan="2">70</td><td colspan="2">75</td><td colspan="2">85</td><td colspan="2">90</td><td colspan="2">95</td><td colspan="2">100</td><td colspan="2">120</td><td colspan="2">125</td><td colspan="2">220</td></tr><tr><td colspan="2">55</td><td colspan="2">65</td><td colspan="2">72</td><td colspan="2">80</td><td colspan="2">87</td><td colspan="2">92</td><td colspan="2">97</td><td colspan="2">110</td><td colspan="2">122</td><td colspan="2">172</td><td colspan="2">230</td></tr><tr><td colspan="2"><= ></td><td colspan="2"><= ></td><td colspan="2"><= ></td><td colspan="2"><= ></td><td colspan="2"><= ></td><td colspan="2"><= ></td><td colspan="2"><= ></td><td colspan="2"><= ></td><td colspan="2"><= ></td><td colspan="2"><= ></td><td colspan="2"><= ></td></tr><tr><td colspan="2">Yes</td><td>0</td><td>3</td><td>0</td><td>3</td><td>0</td><td>3</td><td>0</td><td>3</td><td>1</td><td>2</td><td>2</td><td>1</td><td>3</td><td>0</td><td>3</td><td>0</td><td>3</td><td>0</td><td>3</td><td>0</td><td>3</td><td>0</td></tr><tr><td colspan="2">No</td><td>0</td><td>7</td><td>1</td><td>6</td><td>2</td><td>5</td><td>3</td><td>4</td><td>3</td><td>4</td><td>3</td><td>4</td><td>3</td><td>4</td><td>4</td><td>3</td><td>5</td><td>2</td><td>6</td><td>1</td><td>7</td><td>0</td></tr><tr><td colspan="2">Gini</td><td colspan="2">0.420</td><td colspan="2">0.400</td><td colspan="2">0.375</td><td colspan="2">0.343</td><td colspan="2">0.417</td><td colspan="2">0.400</td><td colspan="2"><u>0.300</u></td><td colspan="2">0.343</td><td colspan="2">0.375</td><td colspan="2">0.400</td><td colspan="2">0.420</td></tr></table>																				No		No		No		Yes		Yes		Yes		No		No		No		No		Annual Income																				60		70		75		85		90		95		100		120		125		220		55		65		72		80		87		92		97		110		122		172		230		<= >		<= >		<= >		<= >		<= >		<= >		<= >		<= >		<= >		<= >		<= >		Yes		0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	3	0	No		0	7	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	7	0	Gini		0.420		0.400		0.375		0.343		0.417		0.400		<u>0.300</u>		0.343		0.375		0.400		0.420	
	No		No		No		Yes		Yes		Yes		No		No		No		No																																																																																																																																																																																		
	Annual Income																																																																																																																																																																																																				
	60		70		75		85		90		95		100		120		125		220																																																																																																																																																																																		
	55		65		72		80		87		92		97		110		122		172		230																																																																																																																																																																																
	<= >		<= >		<= >		<= >		<= >		<= >		<= >		<= >		<= >		<= >		<= >																																																																																																																																																																																
Yes		0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	3	0																																																																																																																																																																														
No		0	7	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	7	0																																																																																																																																																																														
Gini		0.420		0.400		0.375		0.343		0.417		0.400		<u>0.300</u>		0.343		0.375		0.400		0.420																																																																																																																																																																															

Measure of Impurity: Entropy

- I Entropy at a given node t :

$$Entropy(t) = -\sum_j p(j | t) \log p(j | t)$$

(NOTE: $p(j | t)$ is the relative frequency of class j at node t).

- ◆ **Maximum** ($\log n_c$) when records are **equally distributed** among all classes **implying least information**
 - ◆ **Minimum** (0.0) when all records belong to **one class**, **implying most information**
- Entropy based computations are quite similar to the GINI index computations

Computing Entropy of a Single Node

$$Entropy(t) = -\sum_j p(j | t) \log_2 p(j | t)$$

C1	0
C2	6

$$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$$

$$Entropy = -0 \log 0 - 1 \log 1 = -0 - 0 = 0$$

C1	1
C2	5

$$P(C1) = 1/6 \quad P(C2) = 5/6$$

$$Entropy = - (1/6) \log_2 (1/6) - (5/6) \log_2 (5/6) = 0.65$$

C1	2
C2	4

$$P(C1) = 2/6 \quad P(C2) = 4/6$$

$$Entropy = - (2/6) \log_2 (2/6) - (4/6) \log_2 (4/6) = 0.92$$

Computing Information Gain After Splitting

I Information Gain:

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^k \frac{n_i}{n} Entropy(i) \right)$$

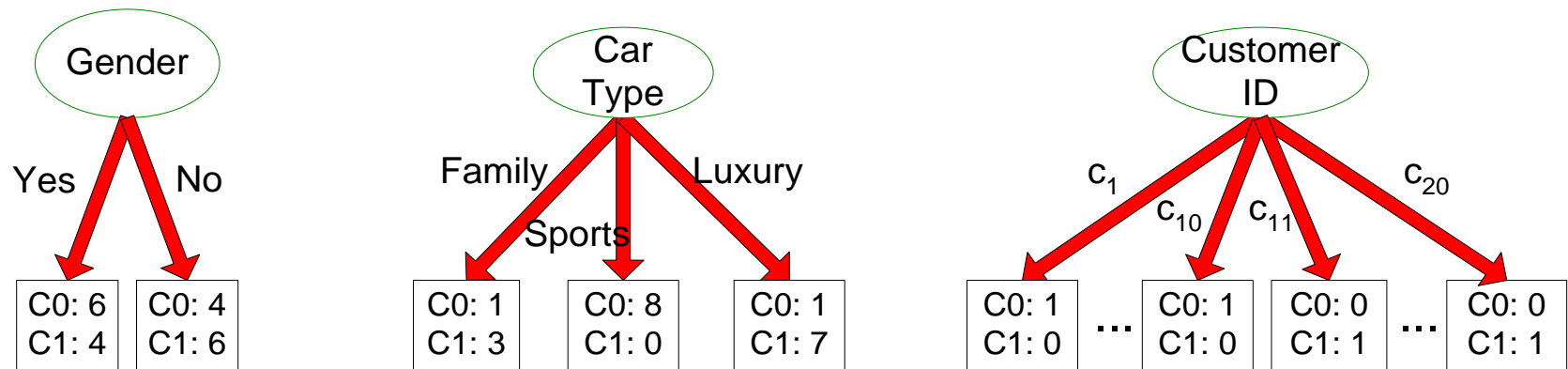
Parent Node, p is split into k partitions;

n_i is number of records in partition i

- Measures **Reduction in Entropy** achieved **because of the split**. Choose the split that achieves most reduction (maximizes GAIN)
- Used in the ID3 and C4.5 decision tree algorithms

Problem with large number of partitions

- «Node impurity» ölçütü, her biri küçük ancak saf olan çok sayıda bölümlle sonuçlanan bölümlmeleri tercih etme eğilimindedir.



- Customer ID has en yüksek bilgi kazancına (highest information gain) sahiptir çünkü tüm çocuklar için entropi sıfırdır

Gain Ratio

| Gain Ratio:

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO} \quad SplitINFO = -\sum_{i=1}^k \frac{n_i}{n} \log \frac{n_i}{n}$$

Parent Node, p is split into k partitions

n_i is the number of records in partition i

- Adjusts Information Gain by the entropy of the partitioning (SplitINFO).
 - ◆ Higher entropy partitioning (large number of small partitions) is penalized!
- Used in C4.5 algorithm
- Designed to overcome the disadvantage of Information Gain

Gain Ratio

| Gain Ratio:

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO} \quad SplitINFO = -\sum_{i=1}^k \frac{n_i}{n} \log \frac{n_i}{n}$$

Parent Node, p is split into k partitions

n_i is the number of records in partition i

	CarType		
	Family	Sports	Luxury
C1	1	8	1
C2	3	0	7
Gini	0.163		

SplitINFO = 1.52

	CarType	
	{Sports, Luxury}	{Family}
C1	9	1
C2	7	3
Gini	0.468	

SplitINFO = 0.72

	CarType	
	{Sports}	{Family, Luxury}
C1	8	2
C2	0	10
Gini	0.167	

SplitINFO = 0.97

$$\text{SplitINFO} = -(16/20) \cdot \log_2(16/20) - (4/20) \cdot \log_2(4/20)$$

$$= 0.72$$

$$\text{SplitINFO} = -\sum_{i=1}^k \frac{n_i}{n} \log \frac{n_i}{n}$$

	CarType		
	Family	Sports	Luxury
C1	1	8	1
C2	3	0	7
Gini	0.163		

SplitINFO = 1.52

	CarType	
	{Sports, Luxury}	{Family}
C1	9	1
C2	7	3
Gini	0.468	

SplitINFO = 0.72

	CarType	
	{Sports}	{Family, Luxury}
C1	8	2
C2	0	10
Gini	0.167	

SplitINFO = 0.97

Measure of Impurity: Classification Error

- | Classification error at a node t :

$$Error(t) = 1 - \max_i P(i | t)$$

- **Maximum** ($1 - 1/n_c$) when records are **equally distributed** among all classes, implying **least interesting information**
- **Minimum** (0) when all records belong to **one class**, implying **most interesting information**

Computing Error of a Single Node

$$Error(t) = 1 - \max_i P(i | t)$$

C1	0
C2	6

$$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$$

$$Error = 1 - \max(0, 1) = 1 - 1 = 0$$

C1	1
C2	5

$$P(C1) = 1/6 \quad P(C2) = 5/6$$

$$Error = 1 - \max(1/6, 5/6) = 1 - 5/6 = 1/6$$

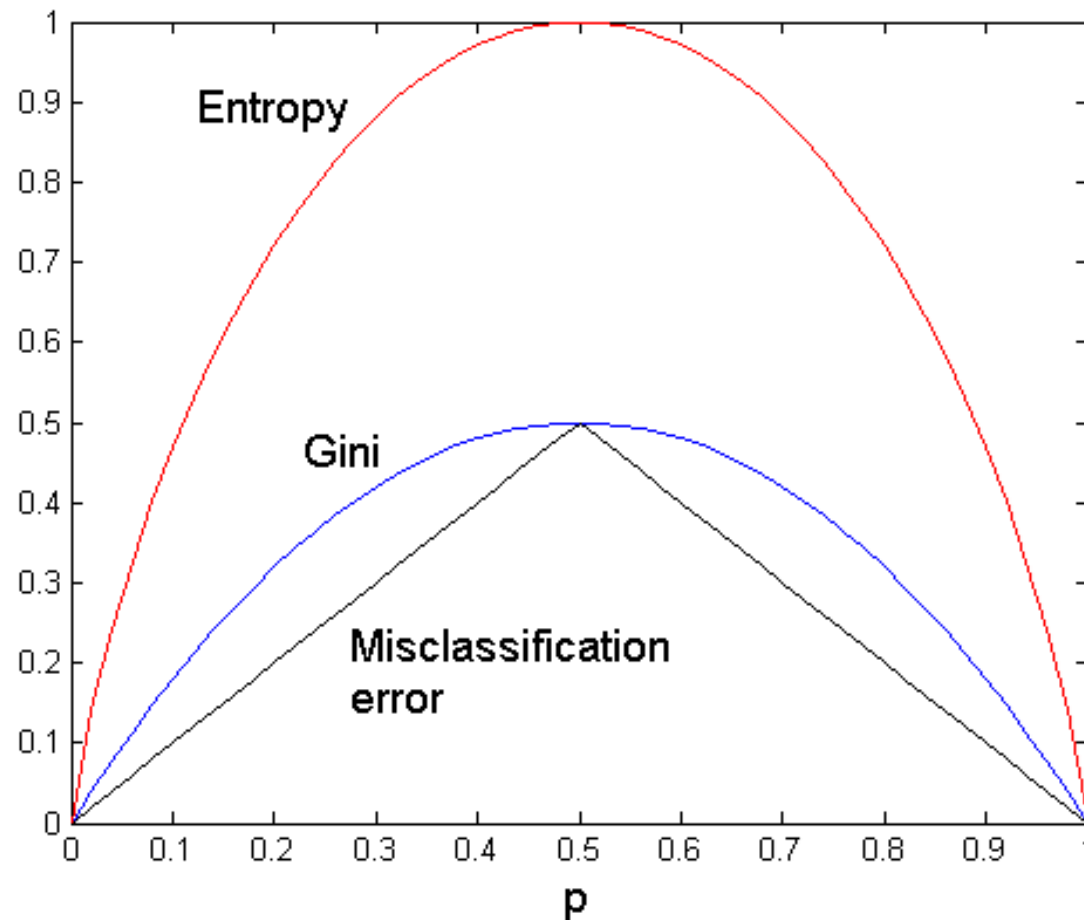
C1	2
C2	4

$$P(C1) = 2/6 \quad P(C2) = 4/6$$

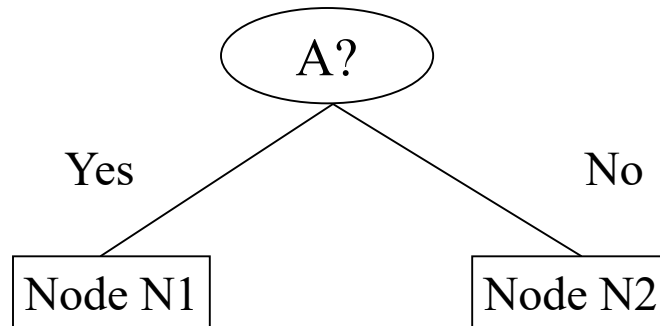
$$Error = 1 - \max(2/6, 4/6) = 1 - 4/6 = 1/3$$

Comparison among Impurity Measures

For a 2-class problem:



Misclassification Error vs Gini Index



	Parent
C1	7
C2	3
Gini = 0.42	

$$\begin{aligned}\text{Gini}(N1) &= 1 - (3/3)^2 - (0/3)^2 \\ &= 0\end{aligned}$$

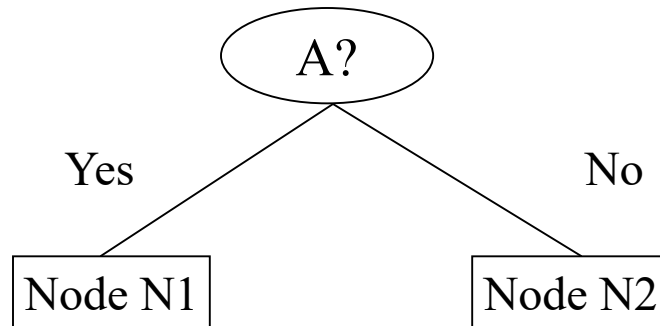
$$\begin{aligned}\text{Gini}(N2) &= 1 - (4/7)^2 - (3/7)^2 \\ &= 0.489\end{aligned}$$

	N1	N2
C1	3	4
C2	0	3
Gini=0.342		

$$\begin{aligned}\text{Gini(Children)} &= 3/10 * 0 \\ &+ 7/10 * 0.489 \\ &= 0.342\end{aligned}$$

**Gini improves but
error remains the
same!!**

Misclassification Error vs Gini Index



	Parent
C1	7
C2	3
Gini = 0.42	

	N1	N2
C1	3	4
C2	0	3
Gini=0.342		

	N1	N2
C1	3	4
C2	1	2
Gini=0.416		

Misclassification error for all three cases = 0.3 !

Decision Tree Based Classification

| Avantajları:

- İnşa etmesi az zahmetlidir
- Bilinmeyen kayıtları (*unknown records*) sınıflandırmada son derece hızlı
- Küçük boyutlu ağaçlar için yorumlanması kolay
- Gürültüye karşı dayanıklı (özellikle overfitting önleme yöntemleri kullanıldığında)
- Gereksiz veya alakasız öznitelikleri kolayca idare edebilir (öznitelikler birbiriyle etkileşim halinde değilse)

| Dezavantajları :

- Olası karar ağacı çözüm uzayı üstel büyüklüktedir. **Greedy** yaklaşımlar çoğu zaman en iyi ağacı bulamaz.
- Öznitelikler arasındaki etkileşimleri hesaba katmaz
- Her karar sınırı yalnızca tek bir özniteliği içerir

Decision Tree Example

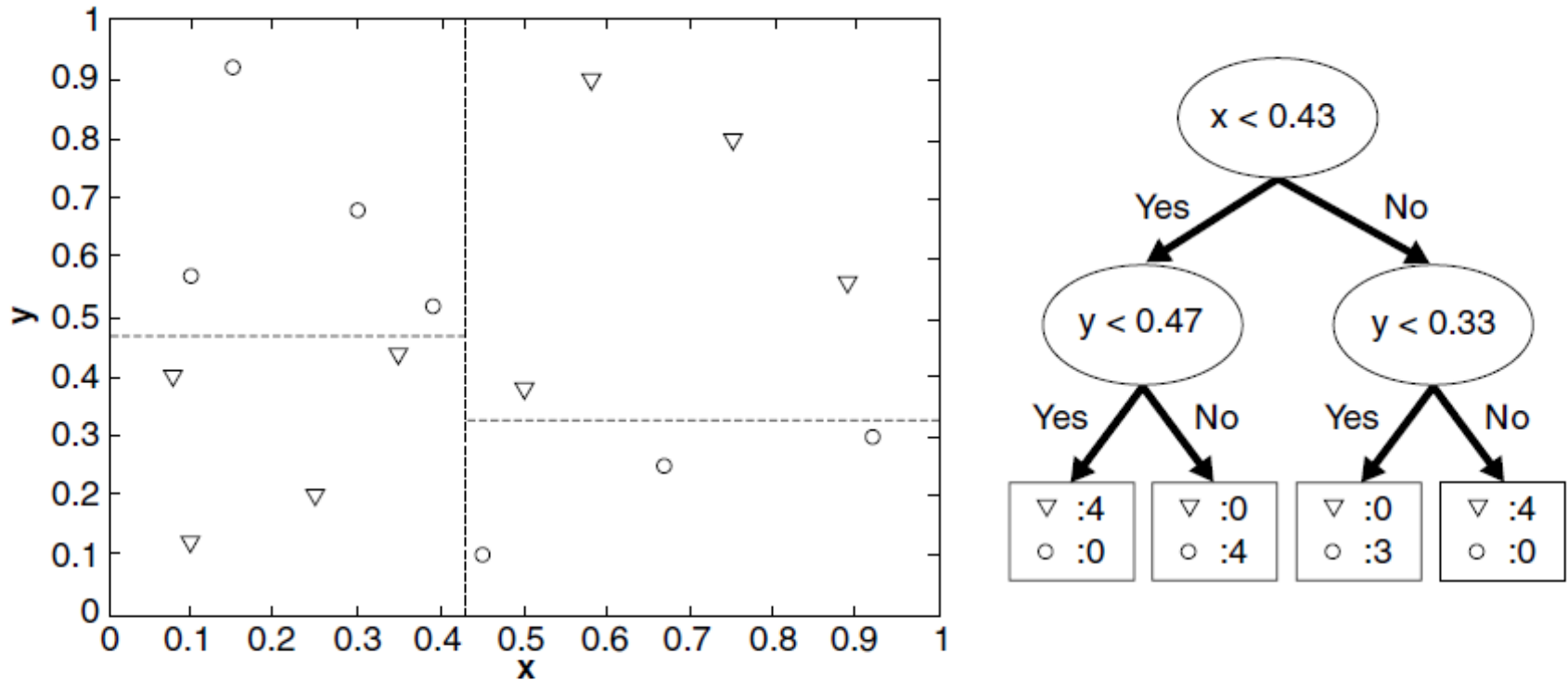


Figure 4.20. Example of a decision tree and its decision boundaries for a two-dimensional data set.

Decision Tree Example

Bu bölümde şimdiye kadar açıklanan test koşulları, bir seferde yalnızca tek bir özniteliğin kullanılmasını içerir. Sonuç olarak, ağaç büyüme prosedürü, her bölge aynı sınıfın kayıtlarını içerene kadar öznitelik uzayını ayrık bölgelere bölme işlemi olarak görülebilir (bkz. Şekil 4.20).

Farklı sınıflardan iki komşu bölge arasındaki sınır, karar sınırı (**decision boundary**) olarak bilinir. Test koşulu yalnızca tek bir özniteliği içerdiğinden, karar sınırları doğrusaldır (**rectilinear**); yani "koordinat eksenlerine" paralel.

Bu, **sürekli özellikler arasındaki karmaşık ilişkileri modellemek** için karar ağacı temsilinin ifade gücünü **sınırlar**. Şekil 4.21, **bir seferde yalnızca tek bir özniteliği** içeren test koşullarını kullanan bir karar ağacı algoritmasıyla etkili bir şekilde sınıflandırılmayan bir veri setini göstermektedir.

Decision Tree Example

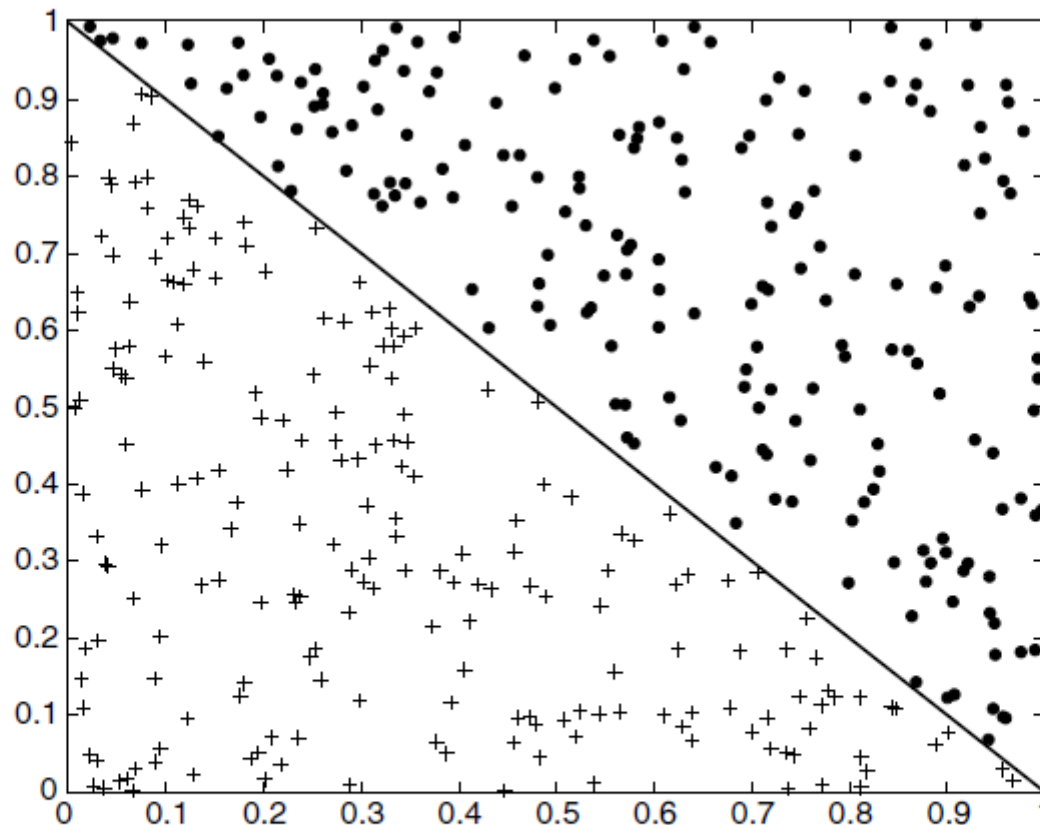


Figure 4.21. Example of data set that cannot be partitioned optimally using test conditions involving single attributes.