Statistics 202: Statistical Aspects of Data Mining

Professor David Mease

Tuesday, Thursday 9:00-10:15 AM Terman 156

Lecture 10 = Start chapter 4

Agenda:

1) Assign 4th Homework (due Tues Aug 7)
2) Start lecturing over Chapter 4 (Sections 4.1-4.5)

Homework Assignment:

Chapter 4 Homework and Chapter 5 Homework Part 1 is due Tuesday 8/7

Either email to me (dmease@stanford.edu), bring it to class, or put it under my office door.

SCPD students may use email or fax or mail.

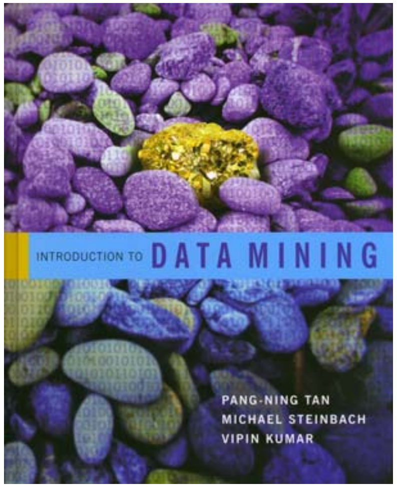
The assignment is posted at

http://www.stats202.com/homework.html

Important: If using email, please submit only a single file (word or pdf) with your name and chapters in the file name. Also, include your name on the first page. Finally, please put your name and the homework # in the subject of the email.

Introduction to Data Mining

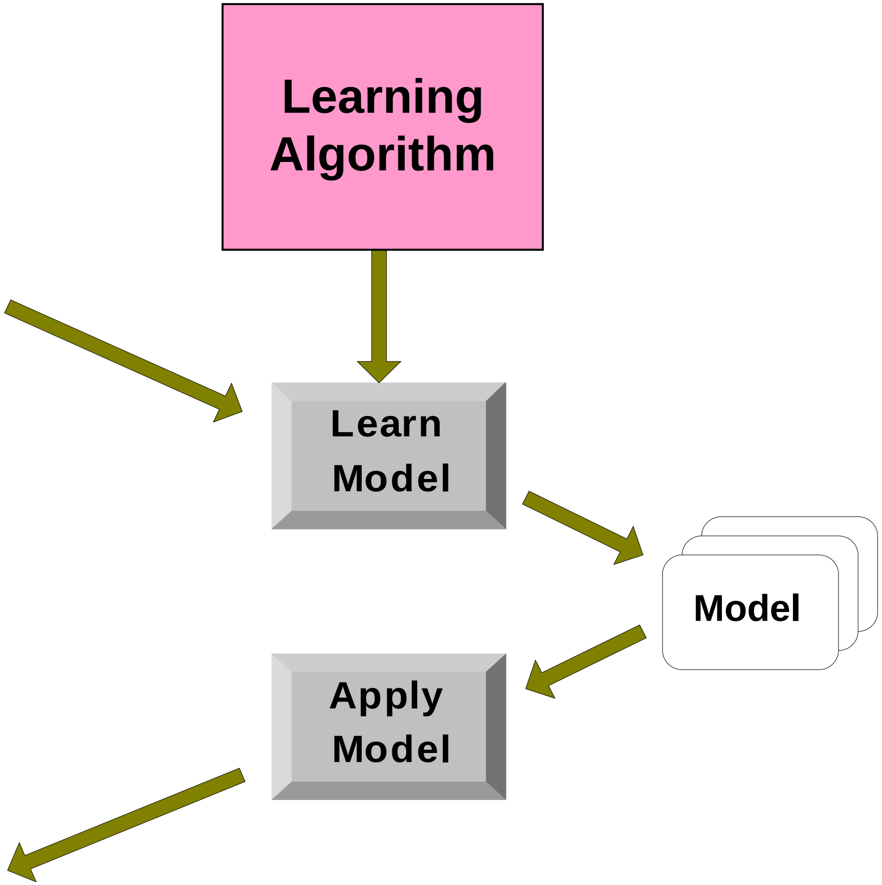
by
Tan, Steinbach, Kumar



Chapter 4: Classification: Basic Concepts, Decision Trees, and Model Evaluation

Illustration of the 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**



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	**?**

Classification: Definition

- Given a collection of records (***training set***)
  - Each record contains a set of ***attributes (x)***, with one additional attribute which is the ***class (y)***.

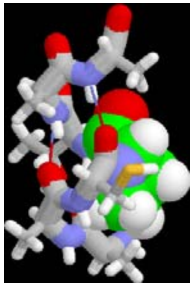
- Find a ***model*** to ***predict*** the class as a function of the values of other attributes.

- Goal: previously unseen records should be assigned a class as accurately as possible.
  - A ***test set*** is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

Classification Examples

- Classifying credit card transactions as legitimate or fraudulent

- Classifying secondary structures of protein as alpha-helix, beta-sheet, or random coil



- Categorizing news stories as finance, weather, entertainment, sports, etc

- Predicting tumor cells as benign or malignant

Classification Techniques

- **There are many techniques/algorithms for carrying out classification**

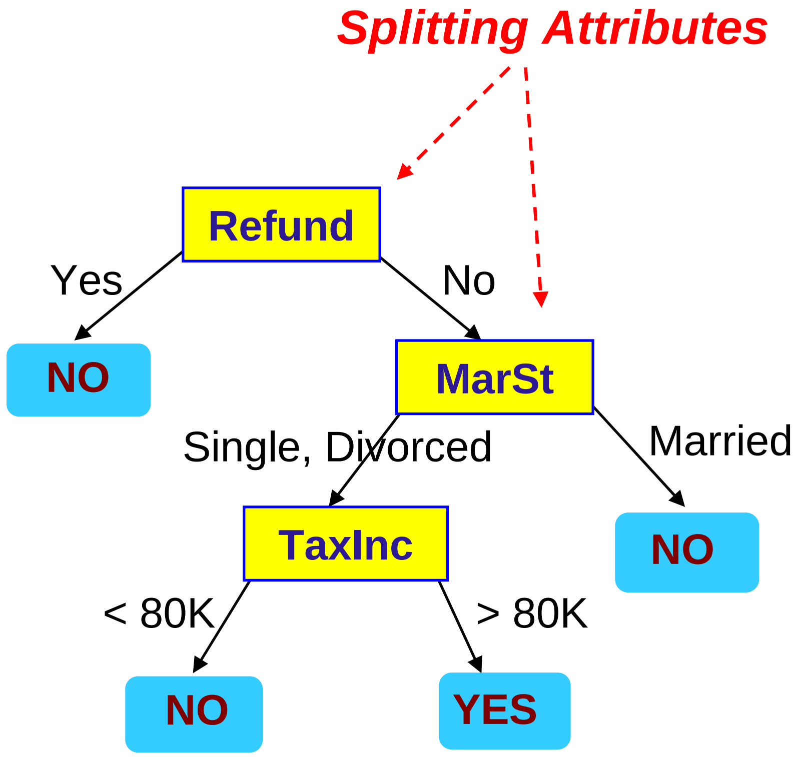
- **In this chapter we will study only *decision trees***

- **In Chapter 5 we will study other techniques, including some very modern and effective techniques**

An Example of a Decision Tree

	categorical	categorical	continuous	class
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
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**

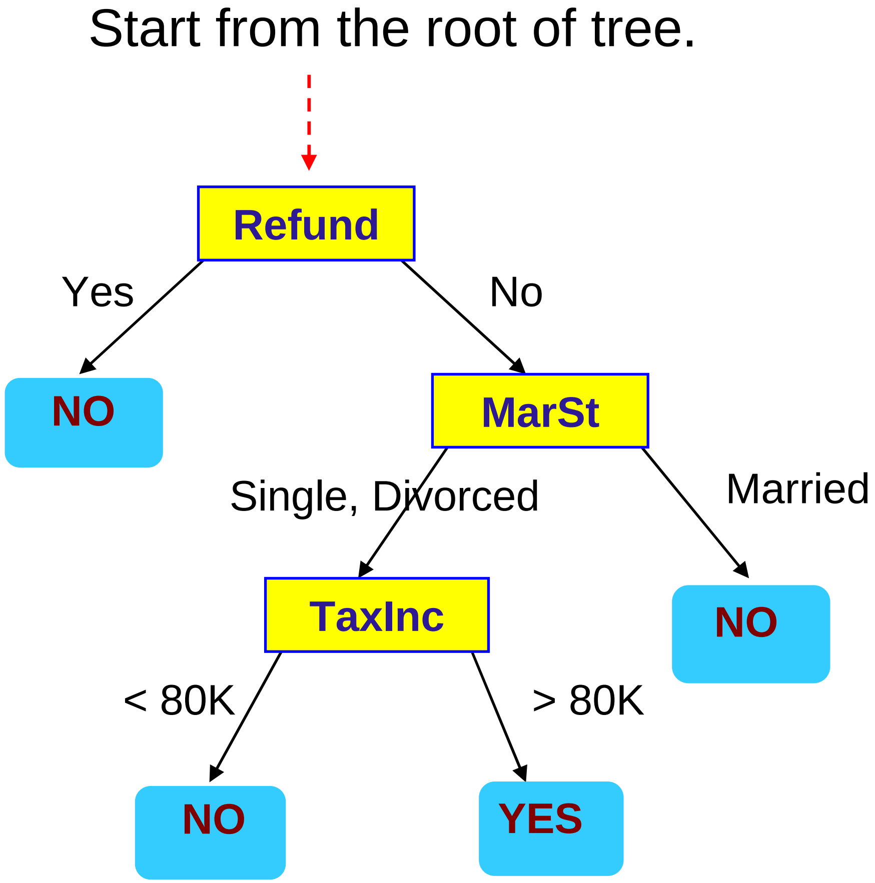
Training Data



Splitting Attributes

Model: Decision Tree

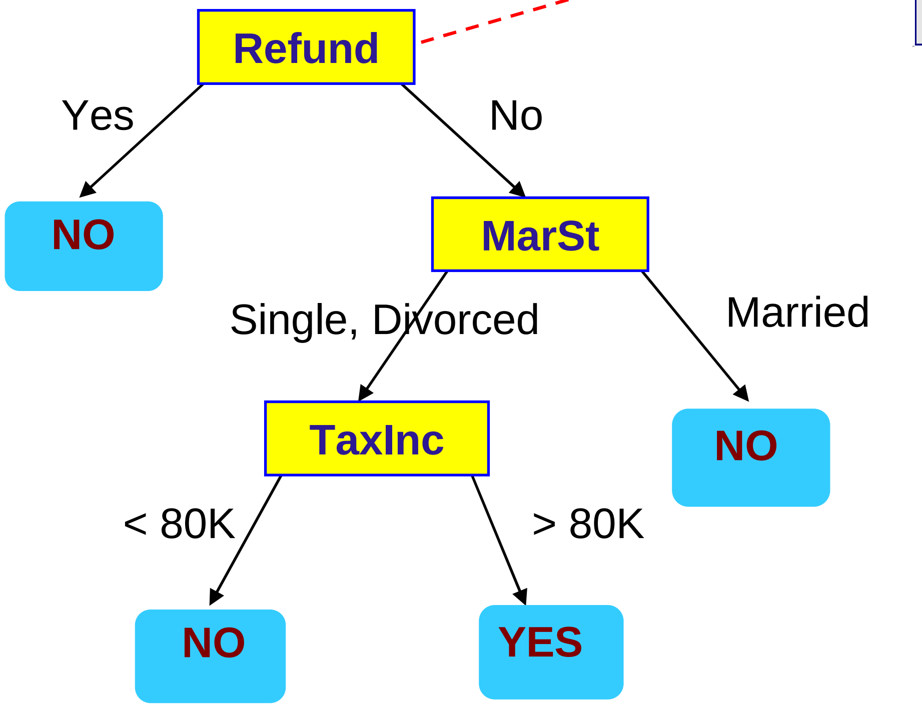
Applying the Tree Model to Predict the Class for a New Observation



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

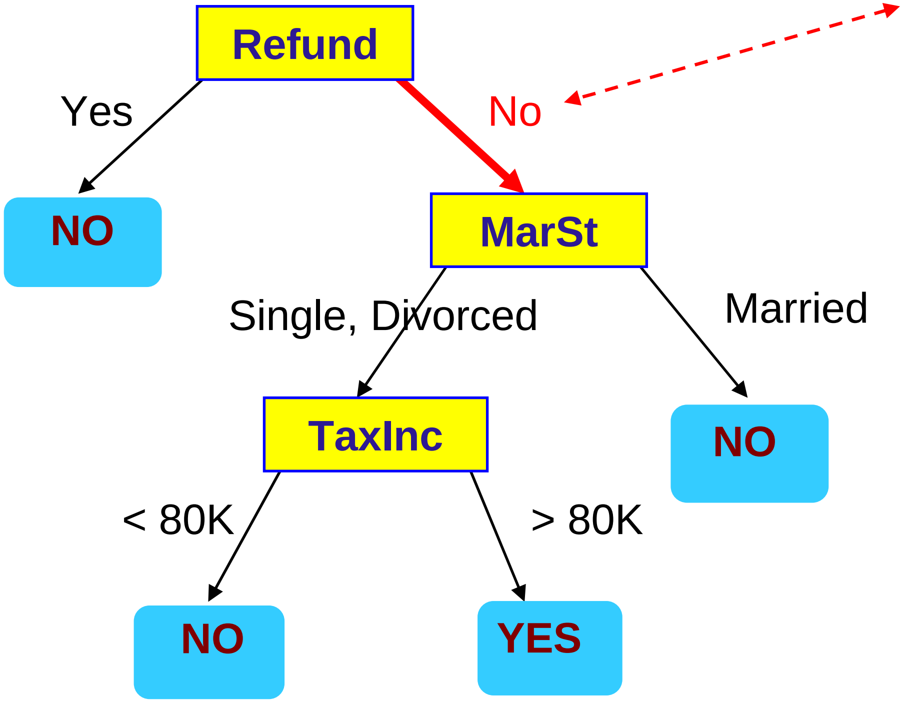
Applying the Tree Model to Predict the Class for a New Observation



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

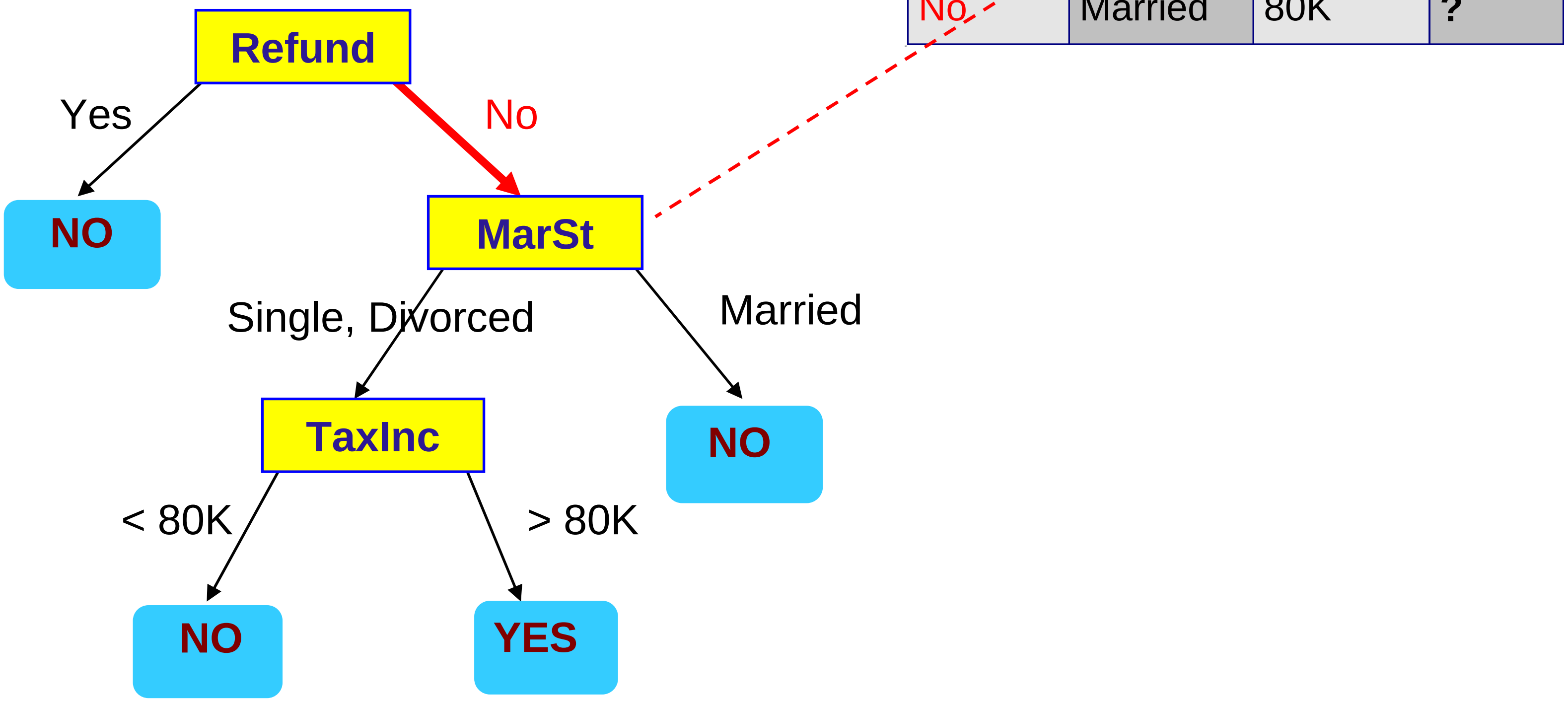
Applying the Tree Model to Predict the Class for a New Observation



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Applying the Tree Model to Predict the Class for a New Observation



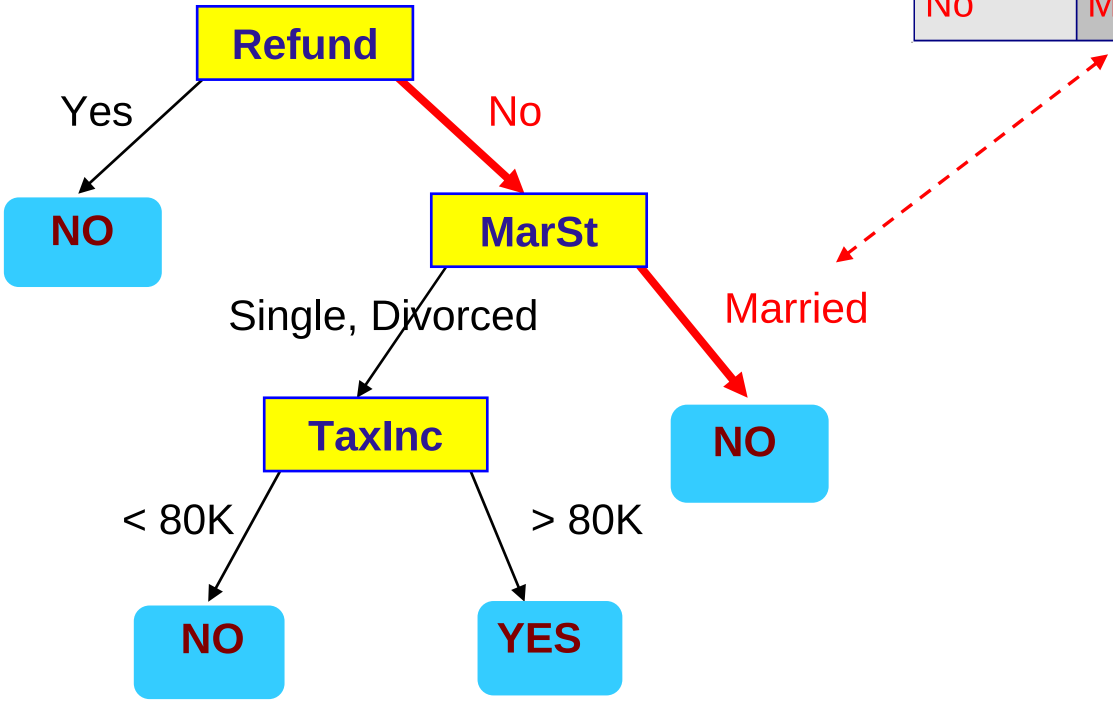
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Applying the Tree Model to Predict the Class for a New Observation

Test Data

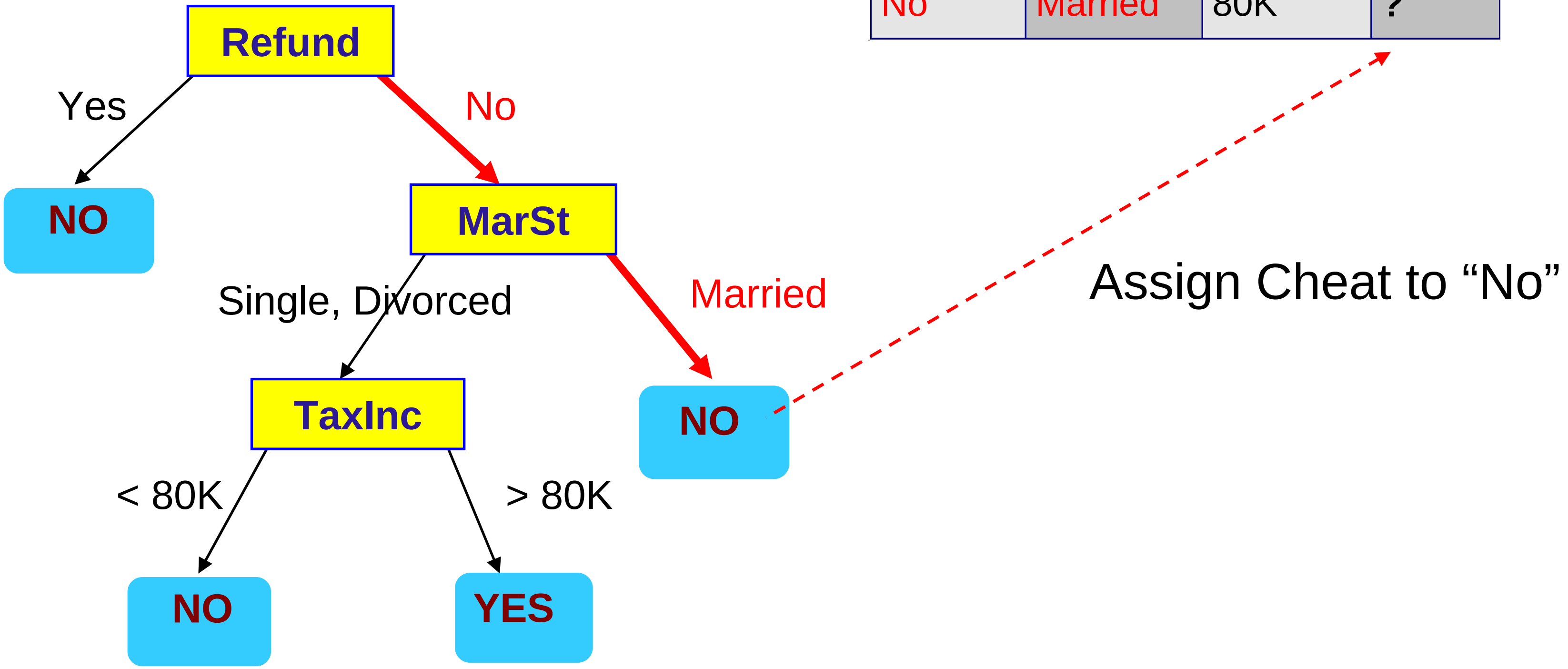
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Applying the Tree Model to Predict the Class for a New Observation

Test Data

Refund	**Marital Status**	**Taxable Income**	**Cheat**
No	Married	80K	**?**



Assign Cheat to “No”

Decision Trees in R

- **The function rpart() in the library “rpart” generates decision trees in R.**

- **Be careful: This function also does *regression trees* which are for a numeric response. Make sure the function rpart() knows your class labels are a factor and not a numeric response.**

(“if y is a factor then method="class" is assumed”)

In class exercise #32:

Below is output from the rpart() function. Use this tree to predict the class of the following observations:

a) (Age=middle Number=5 Start=10)

b) (Age=young Number=2 Start=17)

c) (Age=old Number=10 Start=6)

```
1) root 81 17 absent (0.79012346 0.20987654)
  2) Start>=8.5 62  6 absent (0.90322581 0.09677419)
    4) Age=old,young 48  2 absent (0.95833333 0.04166667)
      8) Start>=13.5 25  0 absent (1.00000000 0.00000000) *
      9) Start< 13.5 23  2 absent (0.91304348 0.08695652) *
    5) Age=middle 14  4 absent (0.71428571 0.28571429)
     10) Start>=12.5 10  1 absent (0.90000000 0.10000000) *
     11) Start< 12.5 4  1 present (0.25000000 0.75000000) *
  3) Start< 8.5 19  8 present (0.42105263 0.57894737)
    6) Start< 4 10  4 absent (0.60000000 0.40000000)
     12) Number< 2.5 1  0 absent (1.00000000 0.00000000) *
     13) Number>=2.5 9  4 absent (0.55555556 0.44444444) *
    7) Start>=4 9  2 present (0.22222222 0.77777778)
     14) Number< 3.5 2  0 absent (1.00000000 0.00000000) *
     15) Number>=3.5 7  0 present (0.00000000 1.00000000) *
```

In class exercise #33:

Use rpart() in R to fit a decision tree to last column of the sonar training data at
http://www-stat.wharton.upenn.edu/~dmease/sonar_train.csv
Use all the default values. Compute the misclassification error on the training data and also on the test data at
http://www-stat.wharton.upenn.edu/~dmease/sonar_test.csv

In class exercise #33:

Use rpart() in R to fit a decision tree to last column of the sonar training data at http://www-stat.wharton.upenn.edu/~dmease/sonar_train.csv Use all the default values. Compute the misclassification error on the training data and also on the test data at http://www-stat.wharton.upenn.edu/~dmease/sonar_test.csv

Solution:

```
install.packages("rpart")
library(rpart)
train<-read.csv("sonar_train.csv",header=FALSE)
y<-as.factor(train[,61])
x<-train[,1:60]
fit<-rpart(y~.,x)
sum(y==predict(fit,x,type="class"))/length(y)
```

In class exercise #33:

Use rpart() in R to fit a decision tree to last column of the sonar training data at
http://www-stat.wharton.upenn.edu/~dmease/sonar_train.csv
Use all the default values. Compute the misclassification error on the training data and also on the test data at
http://www-stat.wharton.upenn.edu/~dmease/sonar_test.csv

Solution (continued):

```
test<-read.csv("sonar_test.csv",header=FALSE)
y_test<-as.factor(test[,61])
x_test<-test[,1:60]
sum(y_test==predict(fit,x_test,type="class"))/
     length(y_test)
```

In class exercise #34:

Repeat the previous exercise for a tree of depth 1 by using control=rpart.control(maxdepth=1). Which model seems better?

In class exercise #34:

Repeat the previous exercise for a tree of depth 1 by using control=rpart.control(maxdepth=1). Which model seems better?

Solution:

```
fit<-
  rpart(y~.,x,control=rpart.control(maxdepth=1))


sum(y==predict(fit,x,type="class"))/length(y)
sum(y_test==predict(fit,x_test,type="class"))/
     length(y_test)
```

In class exercise #35:

Repeat the previous exercise for a tree of depth 6 by using

```
control=rpart.control(minsplit=0,minbucket=0,
          cp=-1,maxcompete=0, maxsurrogate=0,
           usesurrogate=0, xval=0,maxdepth=6)
```

Which model seems better?

In class exercise #35:

Repeat the previous exercise for a tree of depth 6 by using

```
          control=rpart.control(minsplit=0,minbucket=0,
                    cp=-1,maxcompete=0, maxsurrogate=0,
                     usesurrogate=0, xval=0,maxdepth=6)
```

Which model seems better?

Solution:

```
fit<-rpart(y~.,x,
     control=rpart.control(minsplit=0,
          minbucket=0,cp=-1,maxcompete=0,
          maxsurrogate=0, usesurrogate=0,
          xval=0,maxdepth=6))
sum(y==predict(fit,x,type="class"))/length(y)
sum(y_test==predict(fit,x_test,type="class"))/
     length(y_test)
```

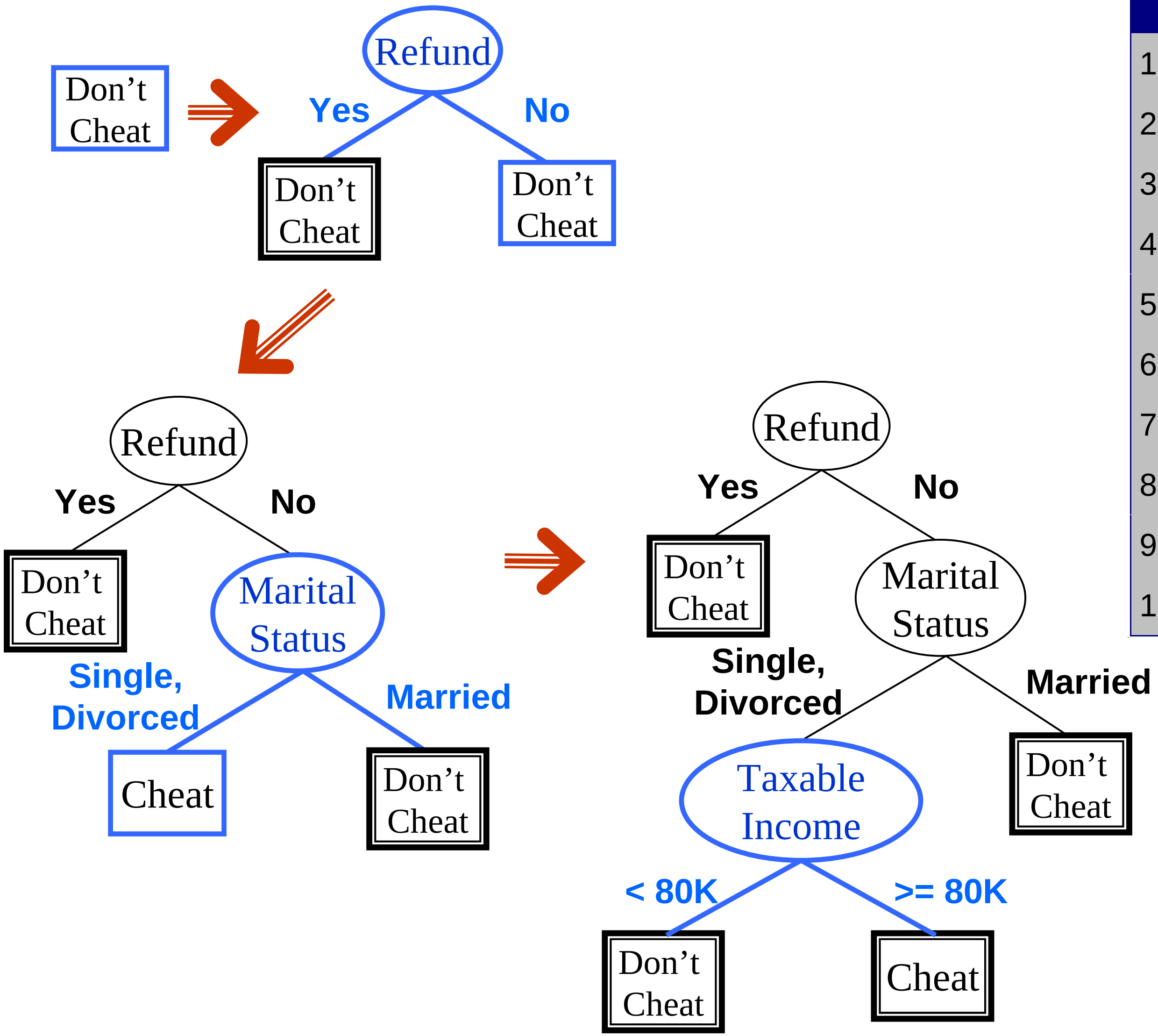
How are Decision Trees Generated?

- Many algorithms use a version of a “top-down” or “divide-and-conquer” approach known as Hunt’s Algorithm (Page 152):

Let D_t be the set of training records that reach a node t

- –If D_t contains records that belong the same class y_t, then t is a leaf node labeled as y_t
- –If D_t contains records that belong to more than one class, use an attribute test to split the data into smaller subsets. Recursively apply the procedure to each subset.

An Example of Hunt’s Algorithm



Tid	Refund	Marital Status	Taxable Income	Cheat
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

How to Apply Hunt's Algorithm

- Usually it is done in a "greedy" fashion.

- "Greedy" means that the optimal split is chosen at each stage according to some criterion.

- This may not be optimal at the end even for the same criterion, as you will see in your homework.

- However, the greedy approach is computational efficient so it is popular.

How to Apply Hunt's Algorithm (continued)

- Using the greedy approach we still have to decide 3 things:

    #1) What attribute test conditions to consider

    #2) What criterion to use to select the "best" split

    #3) When to stop splitting

- For #1 we will consider only binary splits for both numeric and categorical predictors as discussed on the next slide

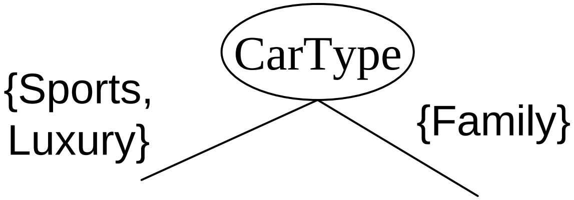
- For #2 we will consider misclassification error, Gini index and entropy

- #3 is a subtle business involving model selection. It is tricky because we don't want to overfit or underfit.

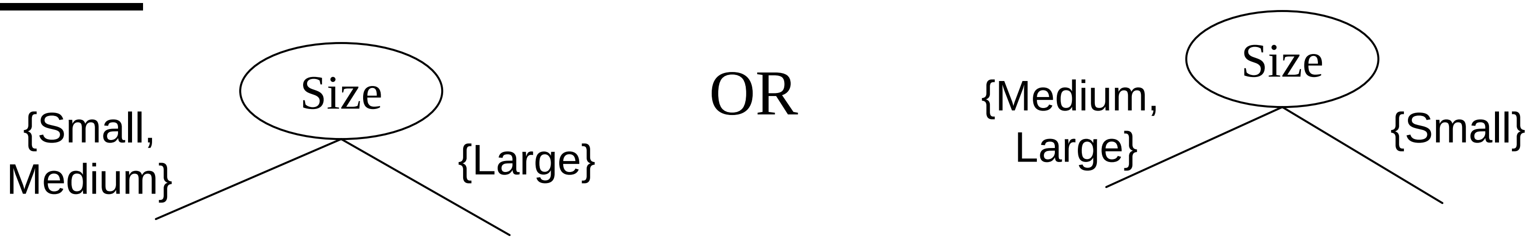
#1) What Attribute Test Conditions to Consider (Section 4.3.3, Page 155)

- We will consider only binary splits for both numeric and categorical predictors as discussed, but your book talks about multiway splits also

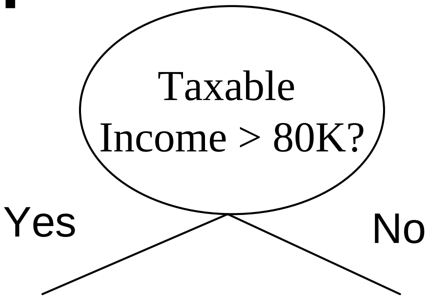
- **Nominal**



- **Ordinal** – like nominal but don’t break order with split



- **Numeric** – often use midpoints between numbers



#2) What criterion to use to select the "best" split (Section 4.3.4, Page 158)

- We will consider misclassification error, Gini index and entropy

Misclassification Error: $\boxed{Error(t) = 1 - \max_i P(i\,|\,t)}$

Gini Index: $\boxed{GINI(t) = 1 - \sum_j [p(j\,|\,t)]^2}$

Entropy: $\boxed{Entropy(t) = -\sum_j p(j\,|\,t)\log_2 p(j\,|\,t)}$

Misclassification Error

$$\boxed{Error(t) = 1 - \max_{i} P(i \mid t)}$$

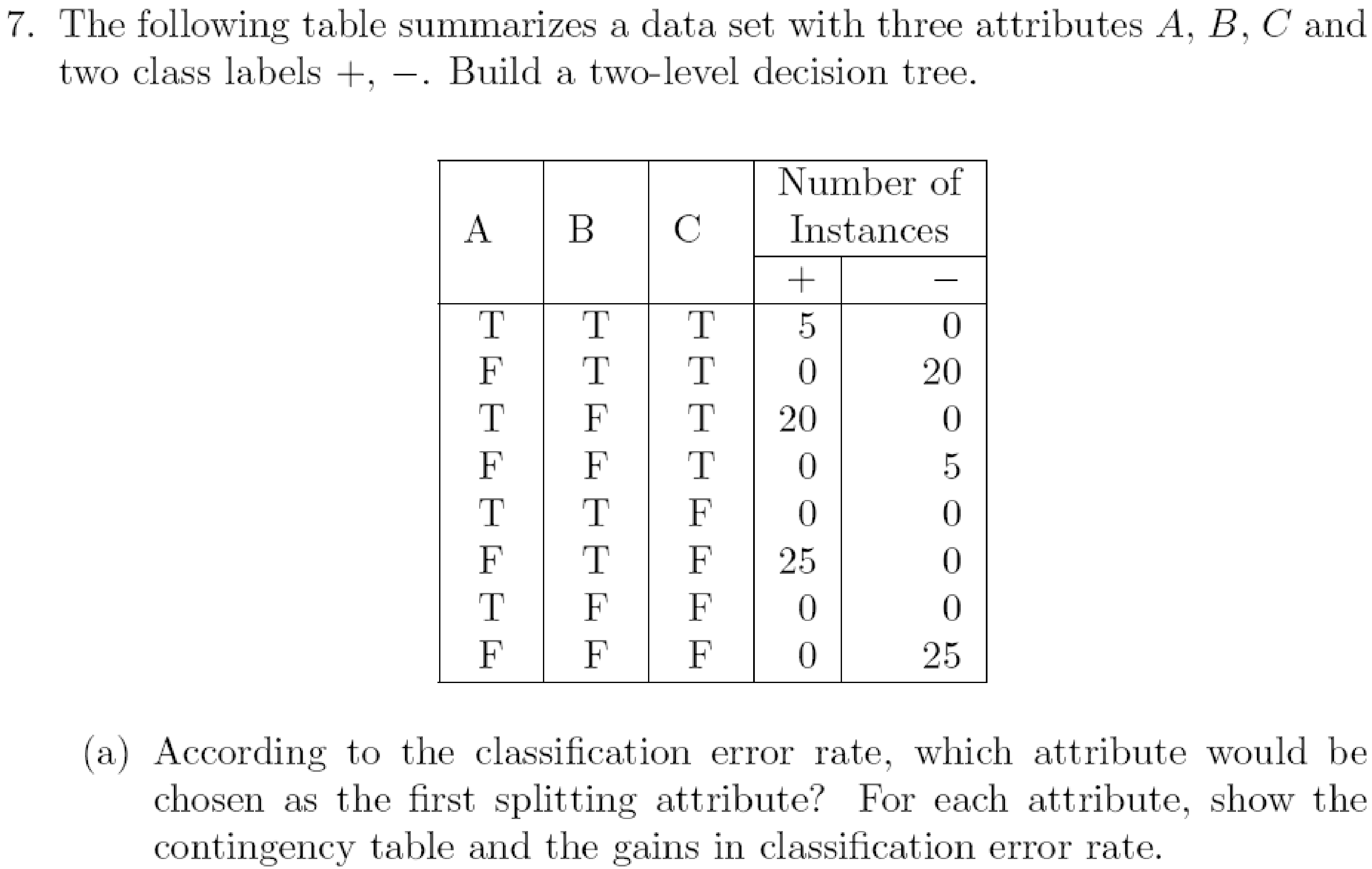
- Misclassification error is usually our final metric which we want to minimize on the test set, so there is a logical argument for using it as the split criterion

- It is simply the fraction of total cases misclassified

- 1 - Misclassification error = “Accuracy” (page 149)

In class exercise #36:

This is textbook question #7 part (a) on page 201.



7. The following table summarizes a data set with three attributes A, B, C and two class labels $+$, $-$. Build a two-level decision tree.

A	B	C	Number of Instances	
			+	−
T	T	T	5	0
F	T	T	0	20
T	F	T	20	0
F	F	T	0	5
T	T	F	0	0
F	T	F	25	0
T	F	F	0	0
F	F	F	0	25

(a) According to the classification error rate, which attribute would be chosen as the first splitting attribute? For each attribute, show the contingency table and the gains in classification error rate.

Gini Index

$$GINI(t) = 1 - \sum_{j} [p(j \mid t)]^2$$

- This is commonly used in many algorithms like CART and the rpart() function in R

- After the Gini index is computed in each node, the overall value of the Gini index is computed as the weighted average of the Gini index in each node

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

Gini Examples for a Single Node

$$\boxed{GINI(t) = 1 - \sum_{j} [p(j \mid t)]^2}$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0   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   P(C2) = 5/6

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

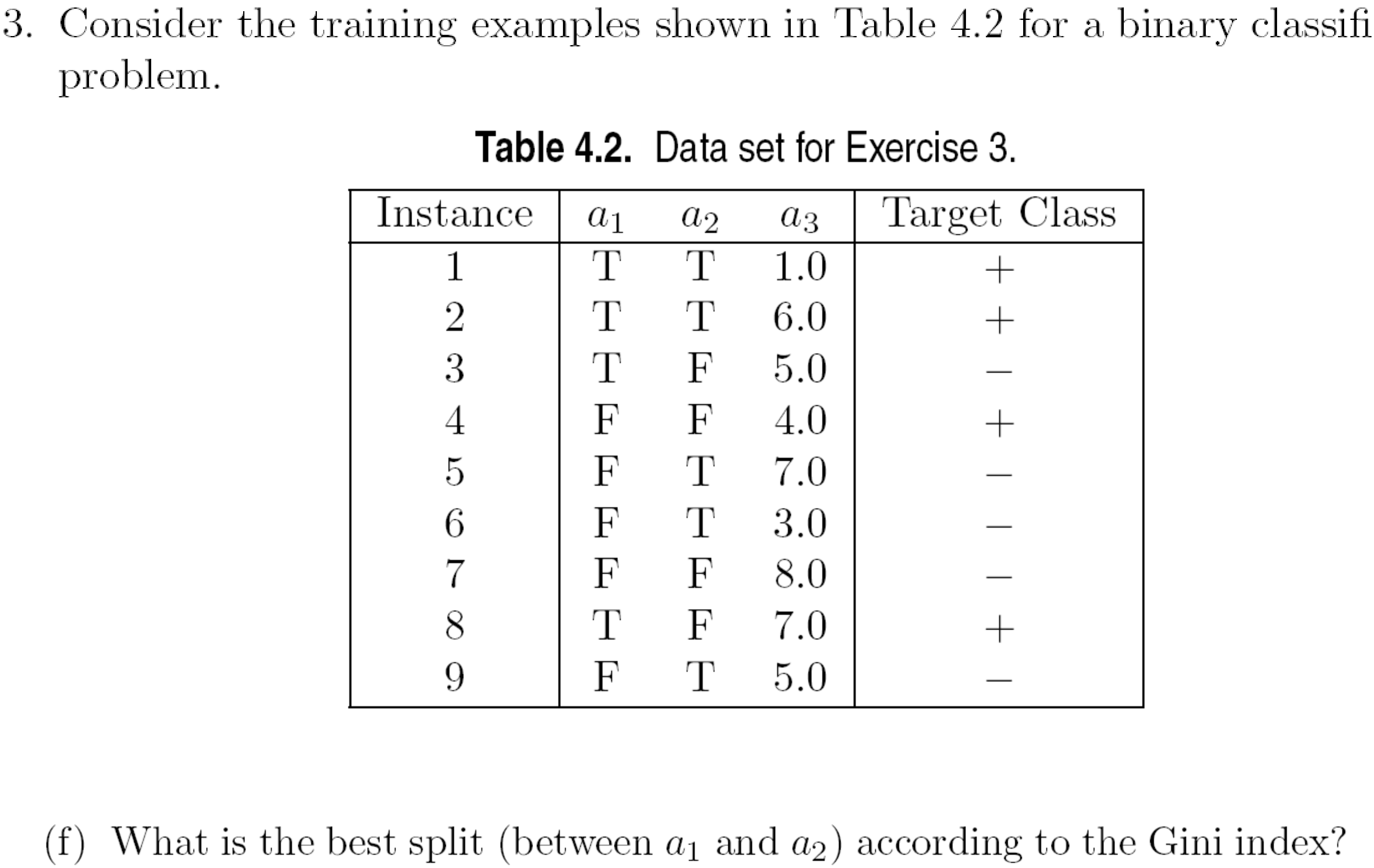
C1	**2**
C2	**4**

P(C1) = 2/6   P(C2) = 4/6

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

In class exercise #37:

This is textbook question #3 part (f) on page 200.



3. Consider the training examples shown in Table 4.2 for a binary classification problem.

Table 4.2. Data set for Exercise 3.

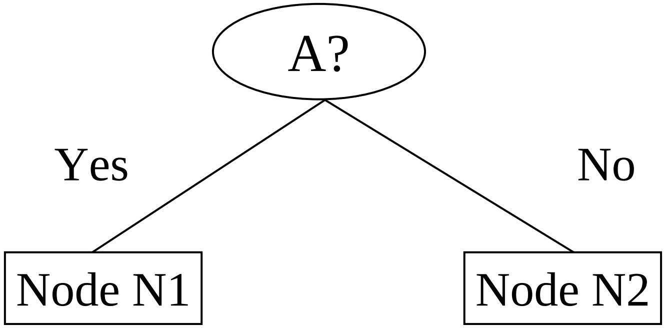
Instance	a_1	a_2	a_3	Target Class
1	T	T	1.0	+
2	T	T	6.0	+
3	T	F	5.0	−
4	F	F	4.0	+
5	F	T	7.0	−
6	F	T	3.0	−
7	F	F	8.0	−
8	T	F	7.0	+
9	F	T	5.0	−

(f) What is the best split (between a_1 and a_2) according to the Gini index?

Misclassification Error Vs. Gini Index

	Parent
C1	7
C2	3
Gini = 0.42	

Gini(N1)
= 1 – $(3/3)^2$ – $(0/3)^2$
= 0



Gini(N2)
= 1 – $(4/7)^2$ – $(3/7)^2$
= 0.490

Gini(Children)
= 3/10 * 0
+ 7/10 * 0.49
= 0.343

	N1	N2
C1	3	4
C2	0	3

- **The Gini index decreases from .42 to .343 while the misclassification error stays at 30%. This illustrates why we often want to use a *surrogate* loss function like the Gini index even if we really only care about misclassification.**

Entropy

$$\boxed{Entropy(t) = -\sum_{j} p(j\,|\,t)\log_2 p(j\,|\,t)}$$

- Measures purity similar to Gini

- Used in C4.5

- After the entropy is computed in each node, the overall value of the entropy is computed as the weighted average of the entropy in each node as with the Gini index

- The decrease in Entropy is called “information gain” (page 160)

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

Entropy Examples for a Single Node

C1	**0**
C2	**6**

P(C1) = 0/6 = 0   P(C2) = 6/6 = 1

Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

C1	**1**
C2	**5**

P(C1) = 1/6   P(C2) = 5/6

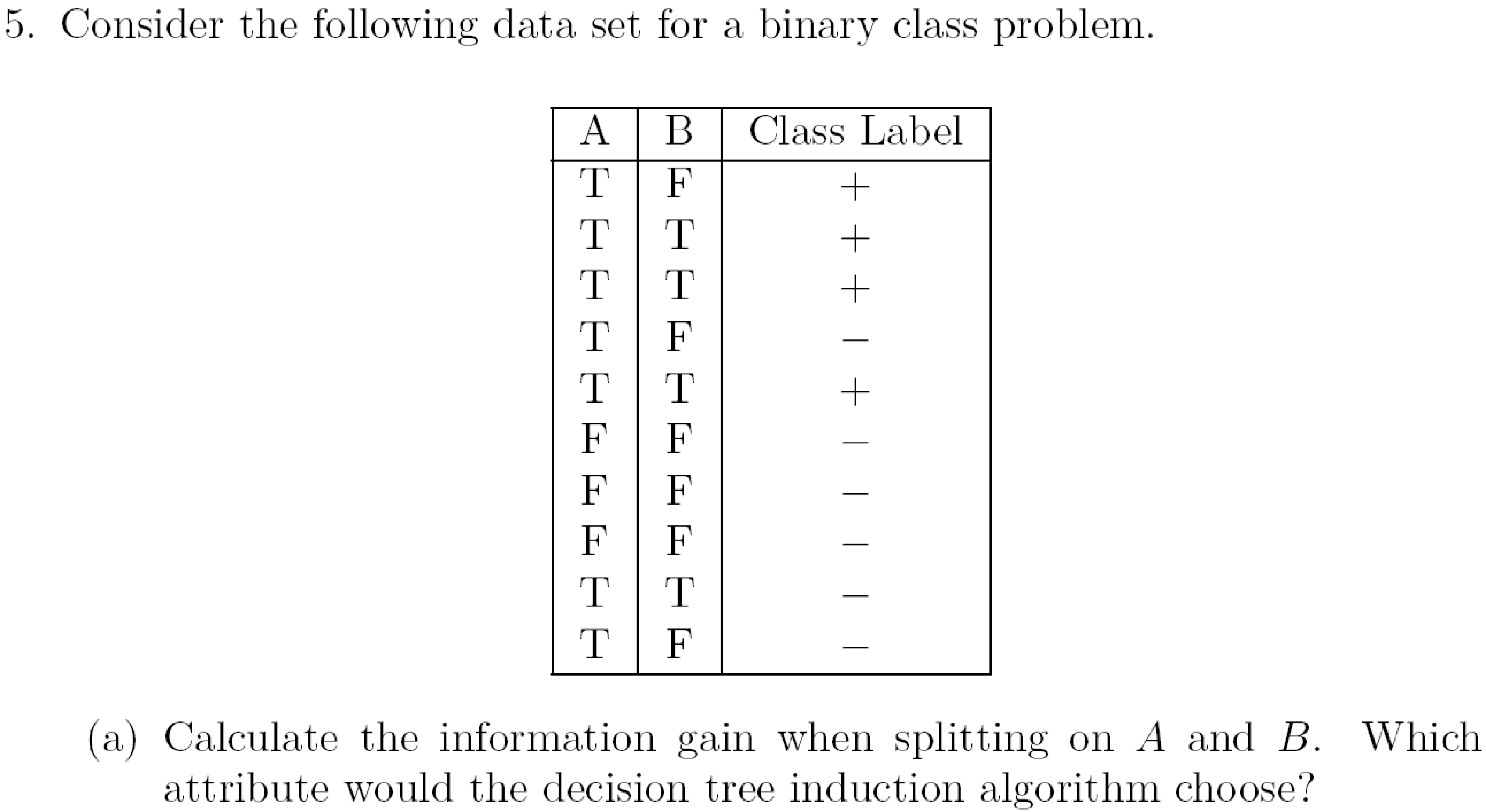
Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (1/6) = 0.65

C1	**2**
C2	**4**

P(C1) = 2/6   P(C2) = 4/6

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

In class exercise #38:
This is textbook question #5 part (a) on page 200.



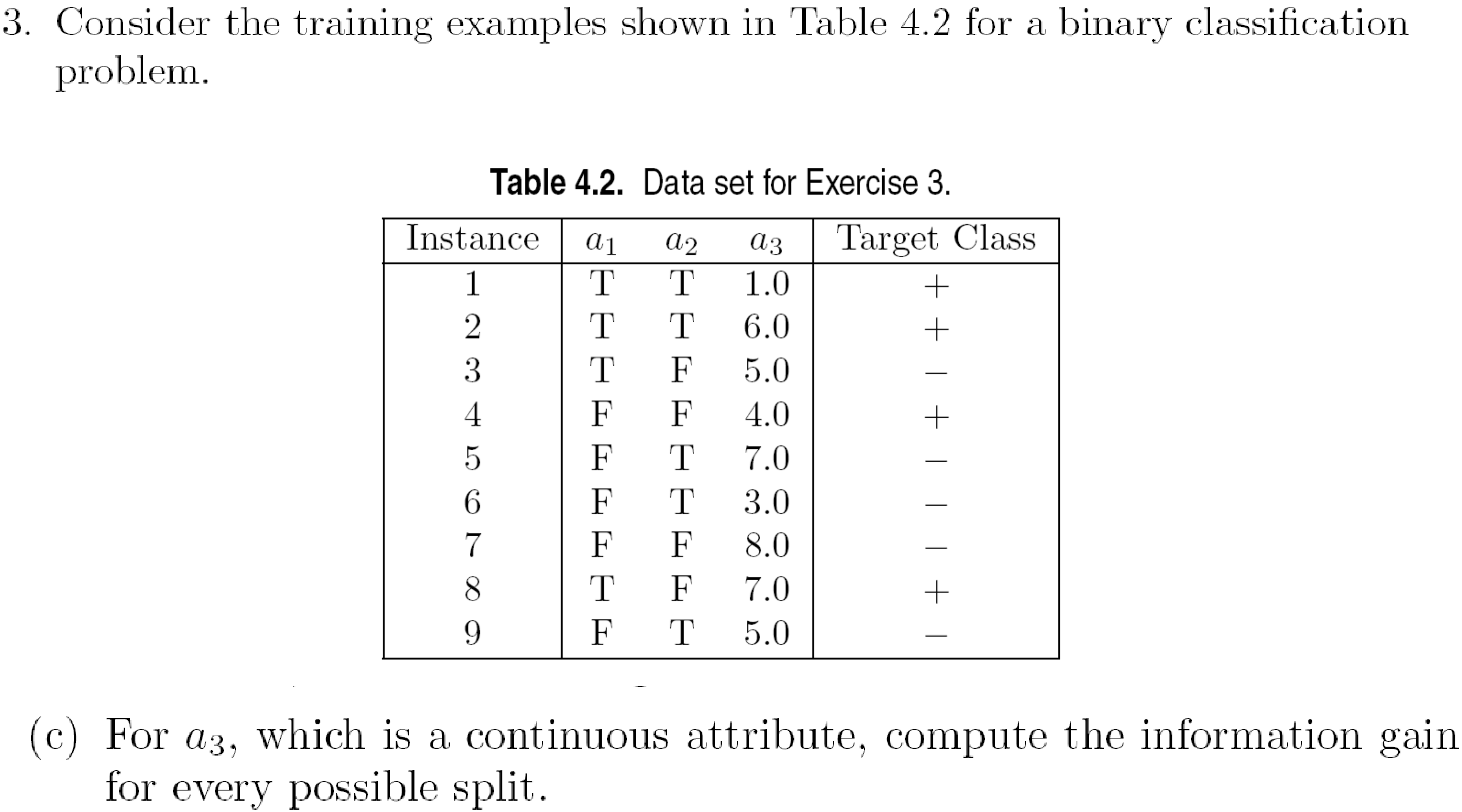
5. Consider the following data set for a binary class problem.

A	B	Class Label
T	F	+
T	T	+
T	T	+
T	F	−
T	T	+
F	F	−
F	F	−
F	F	−
T	T	−
T	F	−

(a) Calculate the information gain when splitting on A and B. Which attribute would the decision tree induction algorithm choose?

In class exercise #39:

This is textbook question #3 part (c) on page 199. It is part of your homework so we will not do all of it in class.



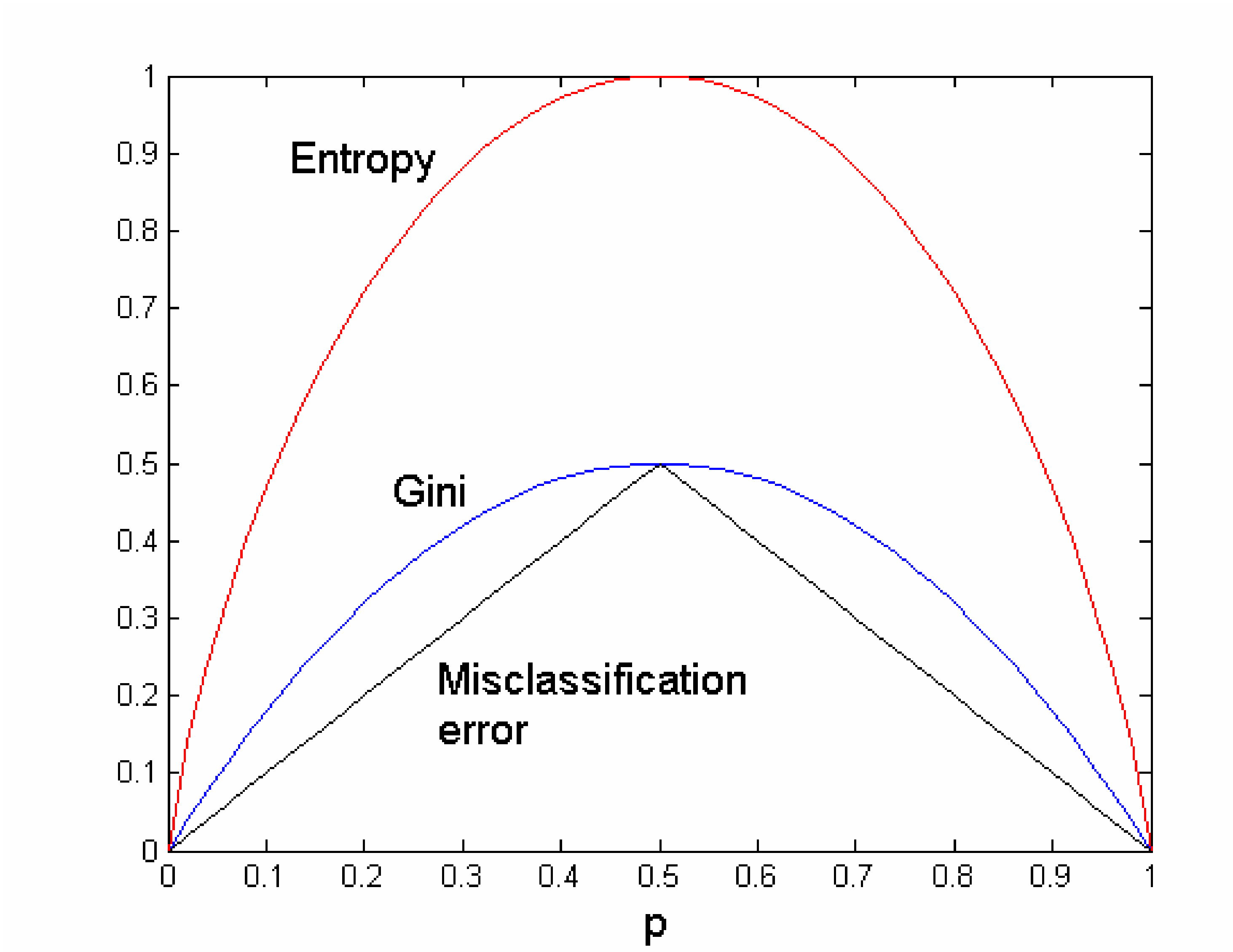
3. Consider the training examples shown in Table 4.2 for a binary classification problem.

Table 4.2. Data set for Exercise 3.

Instance	a_1	a_2	a_3	Target Class
1	T	T	1.0	+
2	T	T	6.0	+
3	T	F	5.0	−
4	F	F	4.0	+
5	F	T	7.0	−
6	F	T	3.0	−
7	F	F	8.0	−
8	T	F	7.0	+
9	F	T	5.0	−

(c) For a_3, which is a continuous attribute, compute the information gain for every possible split.

A Graphical Comparison


Entropy
Gini
Misclassification error
p