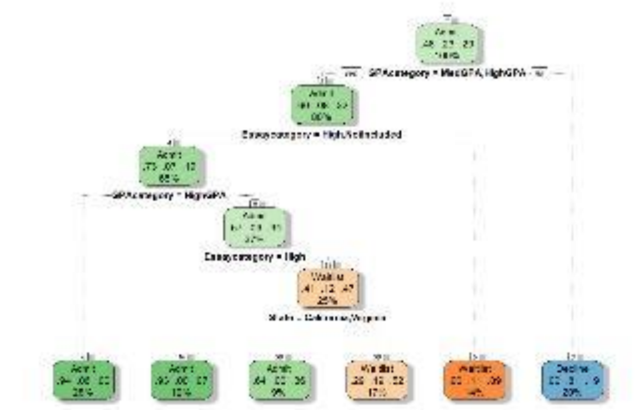
Decision Trees

GATES



Outline

Modeling and Machine Learning part 1

Decision Trees

Instance Based Learning

Good resource book: (**EXTRA READING** ☺)

http://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf

What is a model?

An attempt to **represent reality** through a particular lens.

An **artificial construct** that does not contain unnecessary detail and makes a set of assumptions.

Statistical Modeling

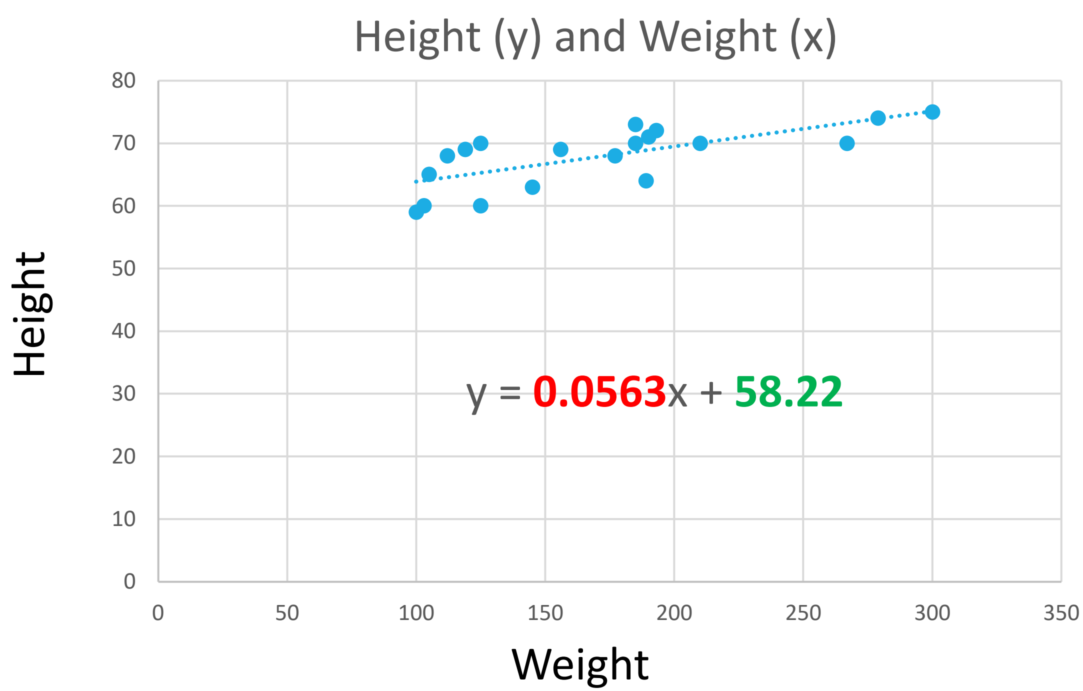
A way to express a **model using mathematics.**

The model designer makes an assumption about the **generative process** of the data – how/where did the data come from?

The goal is to **estimate the parameters** of the model given a particular data set.

A **level of confidence** is always given for the model, e.g. confidence intervals.

Example: Parameter Estimation



Weight	Height
267	70
103	60
193	72
100	59
210	70
189	64
105	65
125	60
156	69
190	71
300	75
119	69
112	68
177	68
145	63
185	70
185	73
279	74
125	70

Using a dataset, we can **estimate the parameters (slope and Y-intercept)** of a linear equation that represents the data (and new data)

Statistical modeling questions

What is the process that generated the data?

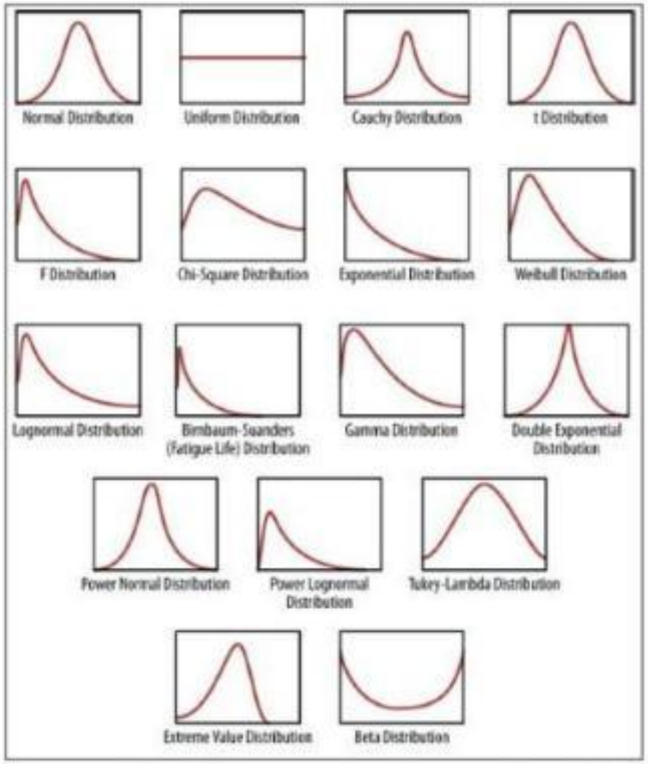
What happened first?

What influences what?

What causes what?

How can I test these?

Common Distributions



Basis for statistical models

Natural processes generate **"shapes" of distributions** that can often be approximated by a mathematical function, given a few parameters that are estimated using the data.

Not all processes generate data that looks like a named distribution.

From book: Doing Data Science

Fitting a Model

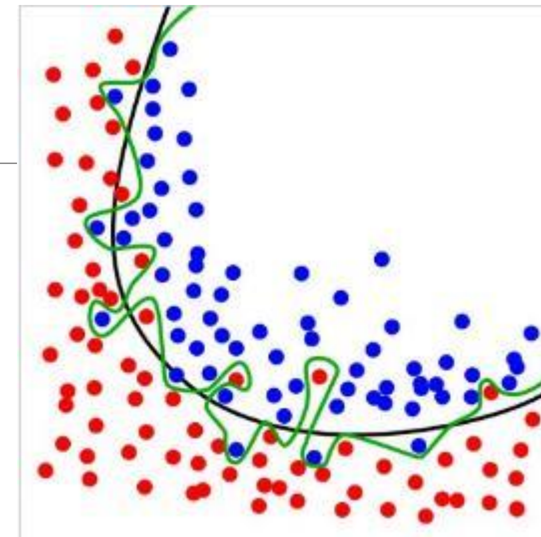
When you **fit a model**, you **estimate its parameters** using real world collected data (samples).

Fitting a model often requires **optimization techniques** and **algorithms**.

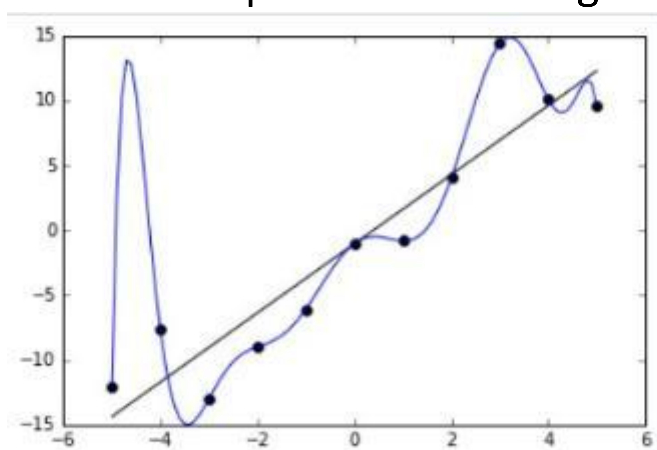
Over fitting is a common problem that needs to be avoided.

- Can end up describing random error or noise rather than the underlying distribution.
- Can occur when a model is too complex.

Avoid testing and training using the same or overlapping data.



Visual Examples of over fitting



Images borrowed from: https://en.wikipedia.org/wiki/Overfitting

Learning Styles

Supervised Learning

- ***Labeled input*** data exist to train a model. The model is then used to predict the class on unseen data.

Unsupervised Learning

- Input data are ***not labeled*** and the result is not known.

Semi-supervised Learning

- Input data is a ***mix*** of labeled and unlabeled examples.

Reinforcement Learning

- A model that **interacts with and learns from its environment.**
- Feedback is provided as ***punishments and rewards in the environment.***

Supervised Examples: Regression, Decision Tree, Random Forest, KNN, Logistic Regression, Naive Bayes, Support Vector Machines, Neural Networks

Unsupervised Examples: kmeans clustering, Association Rules

Reinforcement Learning Examples: Q-Learning, Temporal Difference (TD), Deep Adversarial Networks

Interesting References:
https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

What is Classification

Class == Label ==Group==Category==Cluster

Given: a collection of records/vectors (*training dataset)* Each record contains a set of *attributes (variable values)*, **one of the attributes must be the *class*.**

Goal: Find a *model* (some function of the variable values) to identify the **class of a new vector/record.**

Table 4.1. The vertebrate data set.

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hiber-nates	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 shark	cold-blooded	scales	yes	yes	no	no	no	fish
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

CLASS

Cross-validation: Training and Testing

- Labeled data is required to train a ML model, such as DT, NB, etc.
- A dataset is separated into a TRAINING SET and a TESTING SET.
- The TRAINING SET is used to TRAIN the model – to *teach* the model.
- The TESTING SET is used to determine the accuracy of the model by determining how it predicts data rows (vectors).
- Training sets and testing sets should not overlap in values.
- **Cross-validation** (leave-one-out) is often used.

What Does this Look Like?

Label	Gender	Cholesterol	MaritalStatus	Weight	Height	StressLevel
Risk	M	251	S	267	70	5
NoRisk	F	105	M	103	62	1
Medium	M	156	S	193	72	3
NoRisk	F	109	M	100	63	2
Risk	M	198	S	210	70	4
Risk	F	189	S	189	64	3
NoRisk	F	121	S	105	65	1
Medium	F	134	M	125	60	2
Risk	M	250	S	156	69	5
NoRisk	M	118	M	190	71	3
Risk	F	290	M	300	62	4

Training set

Label	Gender	Cholesterol	MaritalStatus	Weight	Height	StressLevel
Risk	M	251	S	267	70	5
NoRisk	F	105	M	103	62	1
Medium	M	156	S	193	72	3
NoRisk	F	109	M	100	63	2
Risk	M	198	S	210	70	4
Risk	F	189	S	189	64	3
NoRisk	F	121	S	105	65	1

Testing set

?	F	134	M	125	60	2
?	M	250	S	156	69	5
?	M	118	M	190	71	3
?	F	290	M	300	62	4

Example: Feature Table(dataset)

column=feature=attribute=variable=field
row=vector=observation=record=trial

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hiber-nates	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 shark	cold-blooded	scales	yes	yes	no	no	no	fish
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

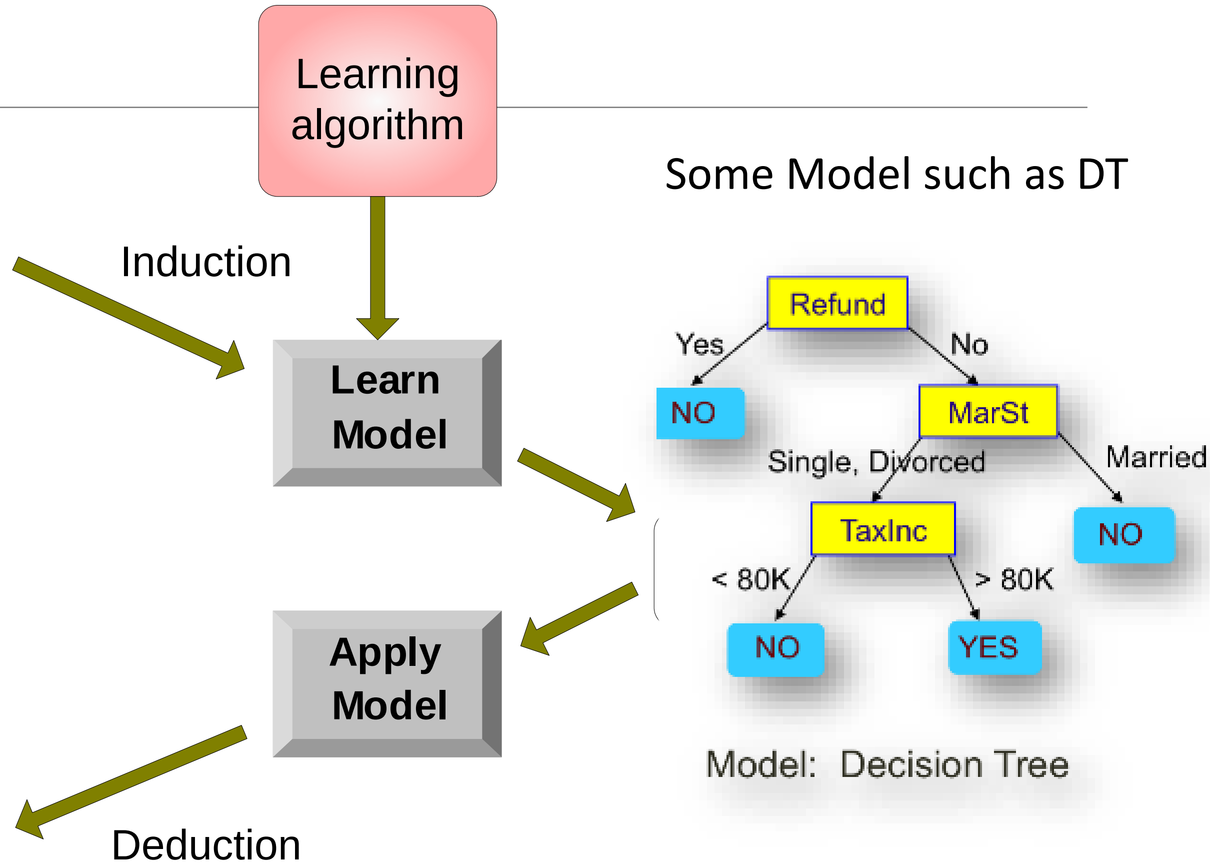
http://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf, page 147

1. **What are the labels?**
2. **What are the data vectors?**

Modeling Data

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	?

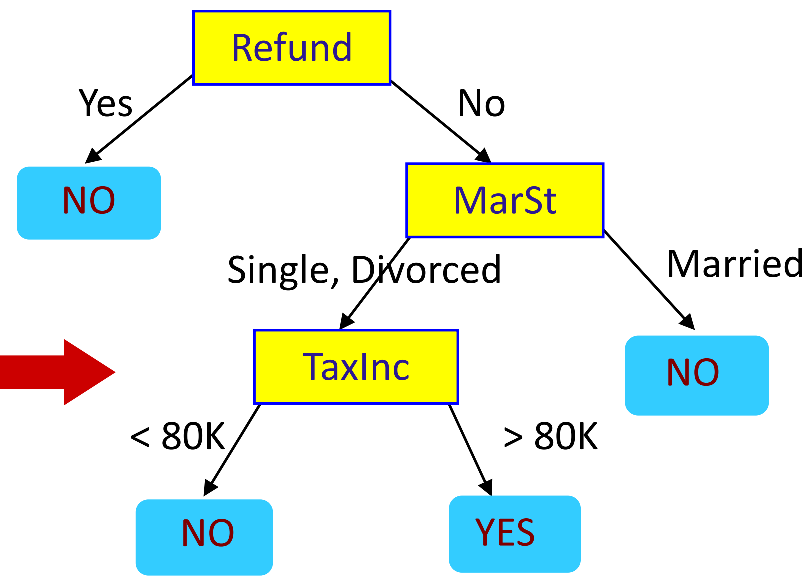
Test Set

Example of Training Data and Decision Tree Model

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**

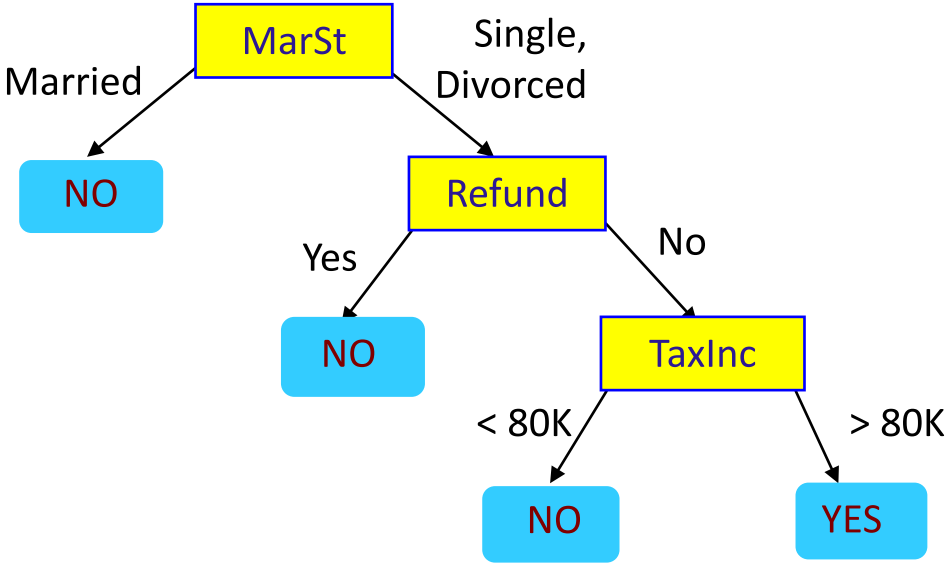
Training Data



Model: Decision Tree

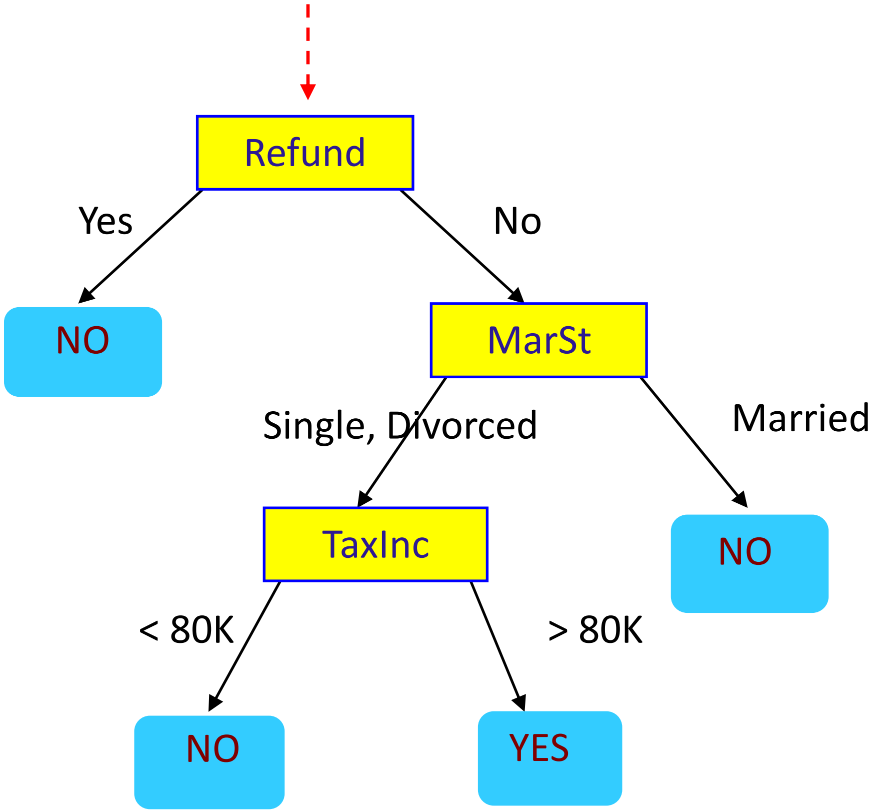
Another Example of Decision Tree – there are infinite tree options

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



Apply Model to Test Data

Start from the root of tree.



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Performance Evaluation: Accuracy and Error Rate

Confusion matrix for a 2-class problem.

		Predicted Class	
		$Class = 1$	$Class = 0$
Actual Class	$Class = 1$	f_{11}	f_{10}
	$Class = 0$	f_{01}	f_{00}

Total Num Correct =
$f_{11} + f_{00}$

The performance of a classification model can be based on **counts** of test records **correctly** or **incorrectly** predicted.

f_{11}: Record was class 1 and was predicted as class 1 correctly
f_{01}: Record was class 0 and incorrectly predicted as Class 1

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}.$$

Equivalently, the performance of a model can be expressed in terms of its **error rate**, which is given by the following equation:

$$\text{Error rate} = \frac{\text{Number of wrong predictions}}{\text{Total number of predictions}} = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}.$$

http://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf, page 5

Metrics for Performance Evaluation: Confusion Matrix

Confusion Matrix:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	True Positive	False Negative
	Class=No	False Positive	True Negative

Example when Accuracy is not a good measure:

Consider a 2-class problem

- Number of Class 0 examples = 9990
- Number of Class 1 examples = 10

If the model predicts everything to be in class 0, accuracy is 9990/10000 = 99.9 %

- Accuracy is misleading because model does not detect any class 1 examples.

Poorly balanced data may not model well.

Using a Cost Matrix

	PREDICTED CLASS		
ACTUAL CLASS	C(i|j)	**Class=Yes**	**Class=No**
	Class=Yes	C(Yes|Yes)	C(No|Yes)
	Class=No	C(Yes|No)	C(No|No)

C(i|j): Cost of **misclassifying** class j, as class i

EXAMPLE: Computing Cost of Classification

This the actual prediction from the model

Model M_1	PREDICTED CLASS		
ACTUAL CLASS		+	-
	+	150	40
	-	60	250

This is the Cost Matrix

Cost Matrix	PREDICTED CLASS		
ACTUAL CLASS	C(i|j)	+	-
	+	-1	100
	-	1	0

Accuracy

= (150+ 250) / (150+40+60+250) = **80%**

Cost

= (150)(-1) + (40)(100) + (60)(1) + (250)(0) = **3910**

Classification Techniques

Decision Tree Methods

Bayesian algorithms (Naïve Bayes)

Support Vector Machines (SVM)

Ensembles (Random Forest)

Decision Trees

ML TOPIC 1

Decision Tree Overview

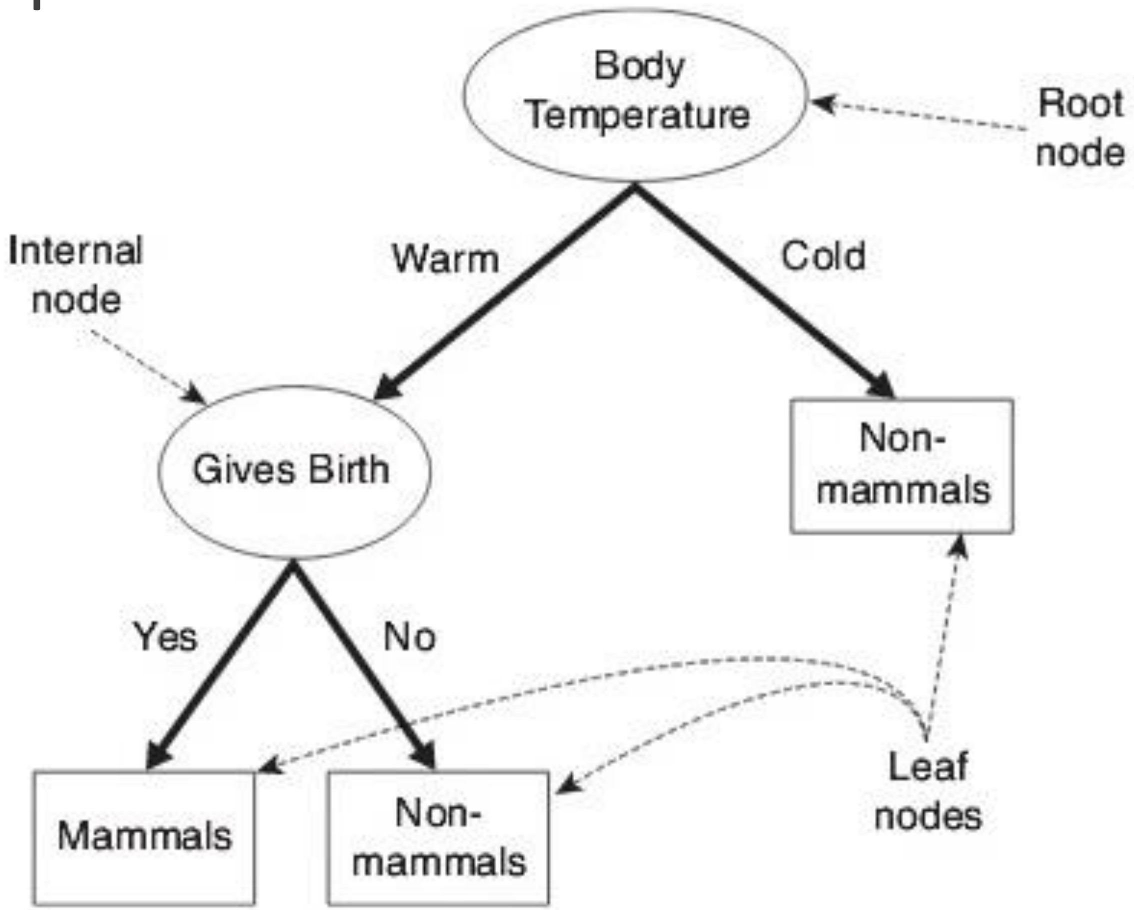
Build a **classifier that is a directional tree structure**.

The tree has

- **Root Node**: no incoming edges and zero or more outgoing edges. (contains attribute test condition(s))
- **Internal Nodes**: Exactly ONE incoming edge and TWO or more outgoing. (contains attribute test condition(s))
- **Leaf/terminal Nodes**: ONE incoming, no outgoing.

Each leaf node is assigned a class label.

Example



http://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf, page 7

Building a Decision Tree

There are an **infinite** number of possible decision trees that can be constructed from a set of attributes.

Finding the **optimal tree** is an **intractable** problem as the search space is exponential.

Algorithms can find "good" decision trees using the **Greedy** approach – they make a series of **locally optimal** decisions.

Example**: Hunt's Algorithm**

- **Hunt is the basis of ID3, C4.5, and CART**

Hunt’s Algorithm: Decision Tree

Assumptions:

Let Dt be the set of **training vectors (rows)**, associated with node t in the tree.

Let **xi** be row (vector) i such that yi is the class label.

All training vectors (also called observations, records, rows) are: [**x**1, **x**2, …, **x**n]with associated class labels [y1, y2, …, yn]

Example:

Risk	M	251	S	267	70	5
NoRisk	F	105	M	103	62	1

Here, our first vector (row) is [M, 251, S, 267, 70, 5] and its label (sometimes called y) is Risk.

Our second vector (row) is [F, 105, M, 103, 62, 1] and its label (sometimes called y) is NoRisk.

Practice: Predict whether a Loan Applicant will repay their loan:

Class label 1
Defaulted = No
Class label 2
Defaulted = Yes

Next, examine a known **training set**.

What are the attributes of each vector (row)?

What is the label of each?

Build a Decision Tree…

binary categorical continuous class

Tid	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

http://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf

GINI and Entropy

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

$$Entropy = \sum_{i=1}^{C} -p_i * \log_2(p_i)$$

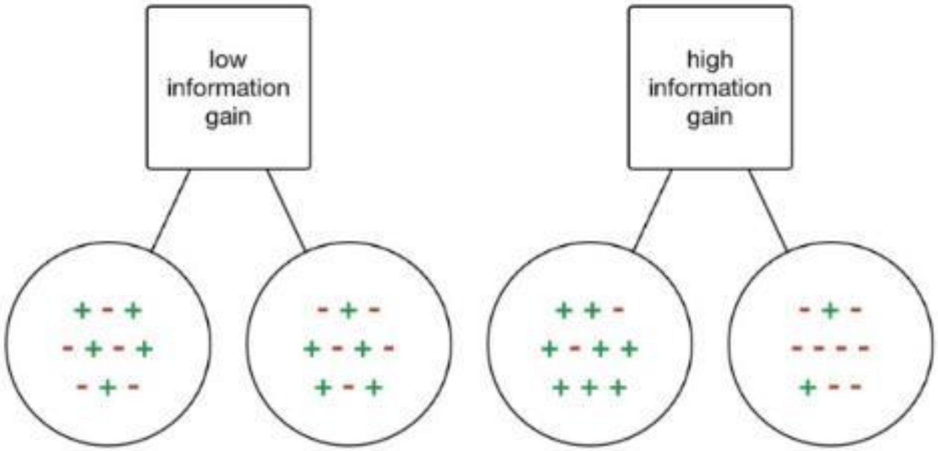
Information Gain

Information gain is a measure quantifies how much a given "tree node split" *unmixes* the labels at a node.

***Mathematically** it is measure of the difference between impurity values **before** splitting the data at a node and the weighted average of the impurity after the split.*

$$Information\ Gain = Entropy(before) - \sum_{j=1}^{K} Entropy(j,\ after)$$

Here – this works using GINI in the same way.



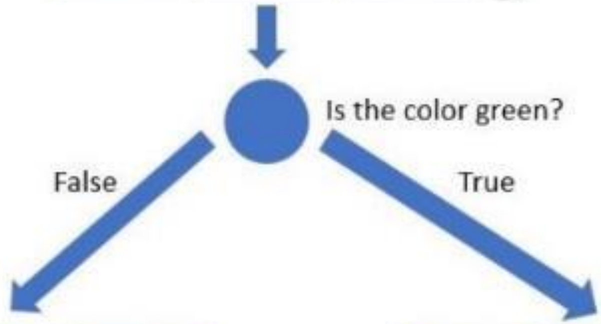
Example 1

GINI:

1 - [(2/5)^2 + (2/5)^2 + (1/5)^2] = .64

Color	Diameter	Label
Green	3	Apple
Yellow	3	Apple
Red	1	Grape
Red	1	Grape
Yellow	3	Lemon

Impurity = 0.64



GINI:

1 - [(1/4)^2 + (2/4)^2 + (1/4)^2] = .625

Impurity = **.625**

Color	Diameter	Label
Yellow	3	Apple
Red	1	Grape
Red	1	Grape
Yellow	3	Lemon

GINI

1 - [(1/1)^2] = 0

Color	Diameter	Label
Green	3	Apple

Impurity = 0

$$Information\ Gain = 0.64 - \left(\frac{4}{5} * \mathbf{.625} + \frac{1}{5} * 0\right) = \mathbf{.14}$$

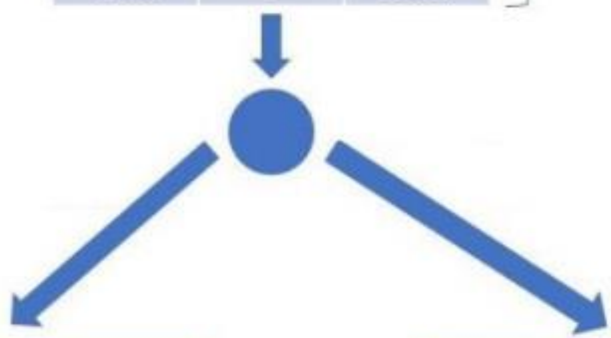
Example 2

Color	Diameter	Label
Green	3	Apple
Yellow	3	Apple
Red	1	Grape
Red	1	Grape
Yellow	3	Lemon

GINI:

$1 - [(2/5)^2 + (2/5)^2 + (1/5)^2] = .64$

Impurity = 0.64



GINI:

$1 - [(2/2)^2] = 0$

Impurity = 0

Color	Diameter	Label
Red	1	Grape
Red	1	Grape

GINI

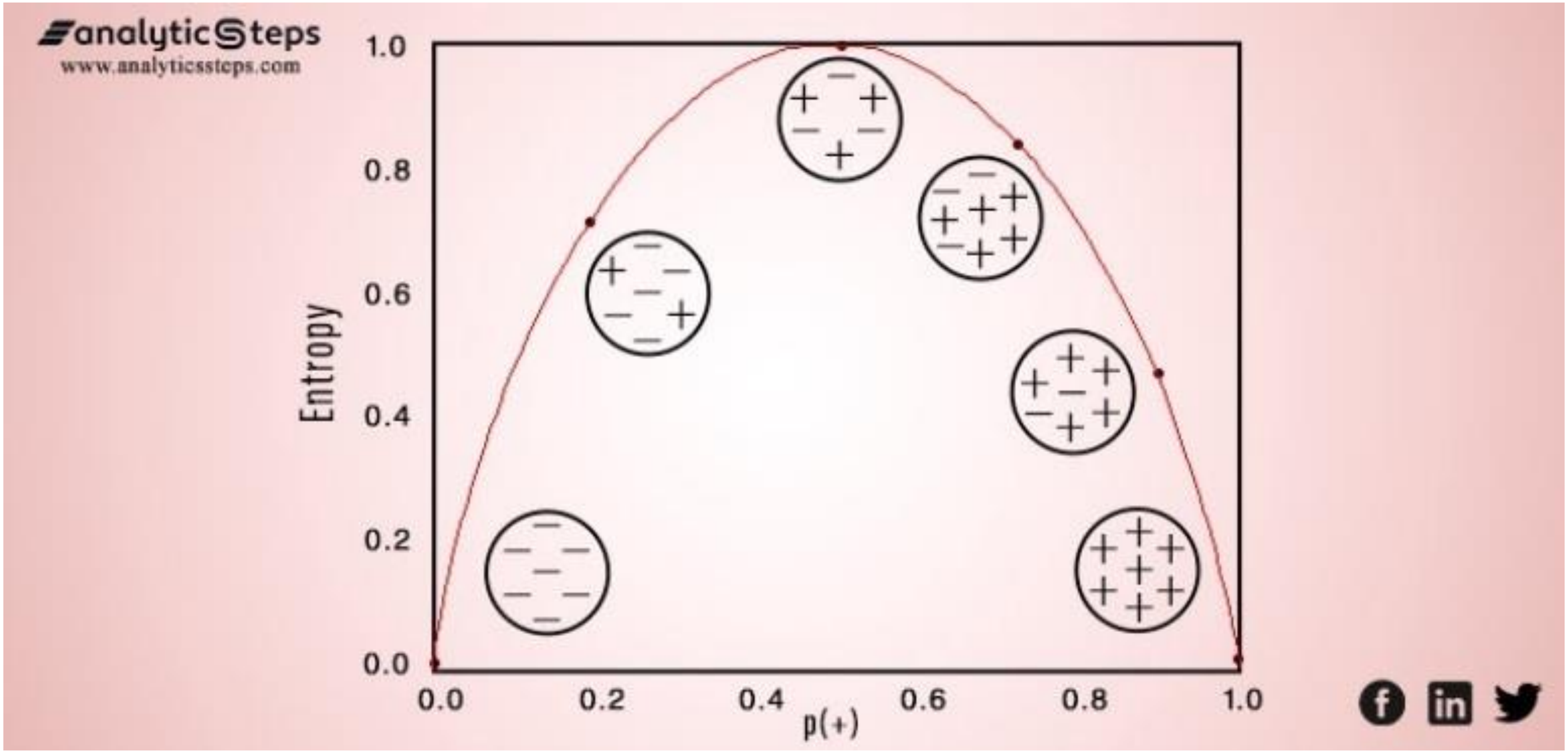
$1 - [(2/3)^2 + (1/3)^2] = .44$

Impurity = .44

Color	Diameter	Label
Green	3	Apple
Yellow	3	Apple
Yellow	3	Lemon

$$Information\ Gain = 0.64 - \left(\frac{2}{5} * 0 + \frac{3}{5} * .44\right) = .376$$

Entropy Visually



Step 1: All data is in the root.
Is the node pure?
If no – split it smartly…

Defaulted = No

C1(no): 7
C2 (yes): 3

Recall that we have two categories:
C1: No
C2: Yes

The initial tree is a single node that represents the fact that most borrows did pay and so the majority is default=no.

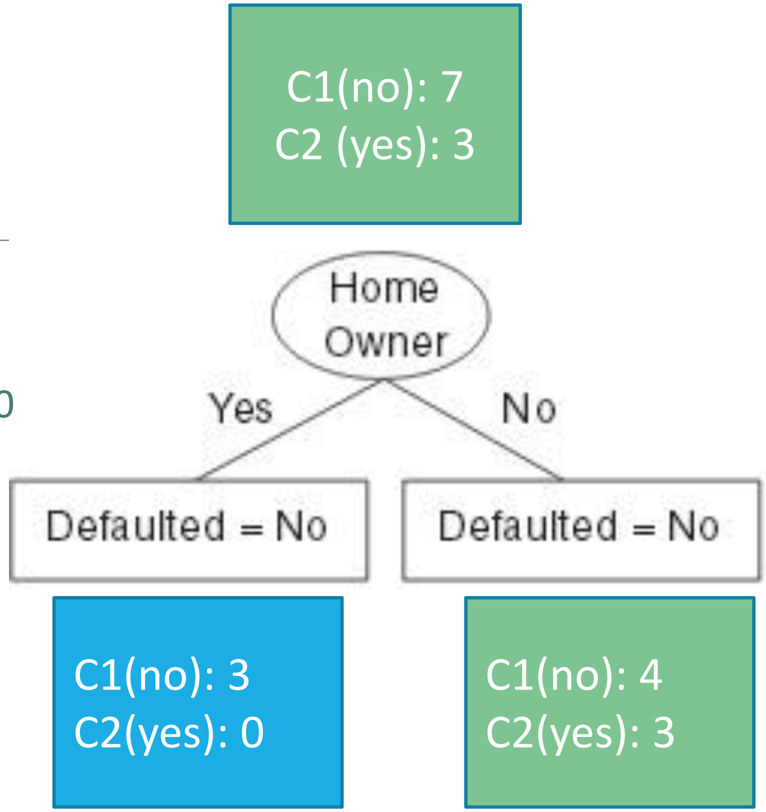
However, **this current node contains records from both classes** and so **must be further refined (split)**.

What is the GINI and Entropy of this node?

GINI = 1 – (7/10)^2 - (3/10)^2 = .42 [0 is pure!] 1 – 0 – 1 = 1 – 1 = 0 or 1 – 1 – 0 = 0

Entropy = – (7/10)$\log_2$(7/10) - (3/10)$\log_2$(3/10) = .88 (0 is pure.)

Step 2



GINI = 1 – (3/3)^2 - (0/3)^2 = 0 (pure)

Entropy = -(3/3)log2(3/3) - (0/3)log2(0/3) = 0 (pure)

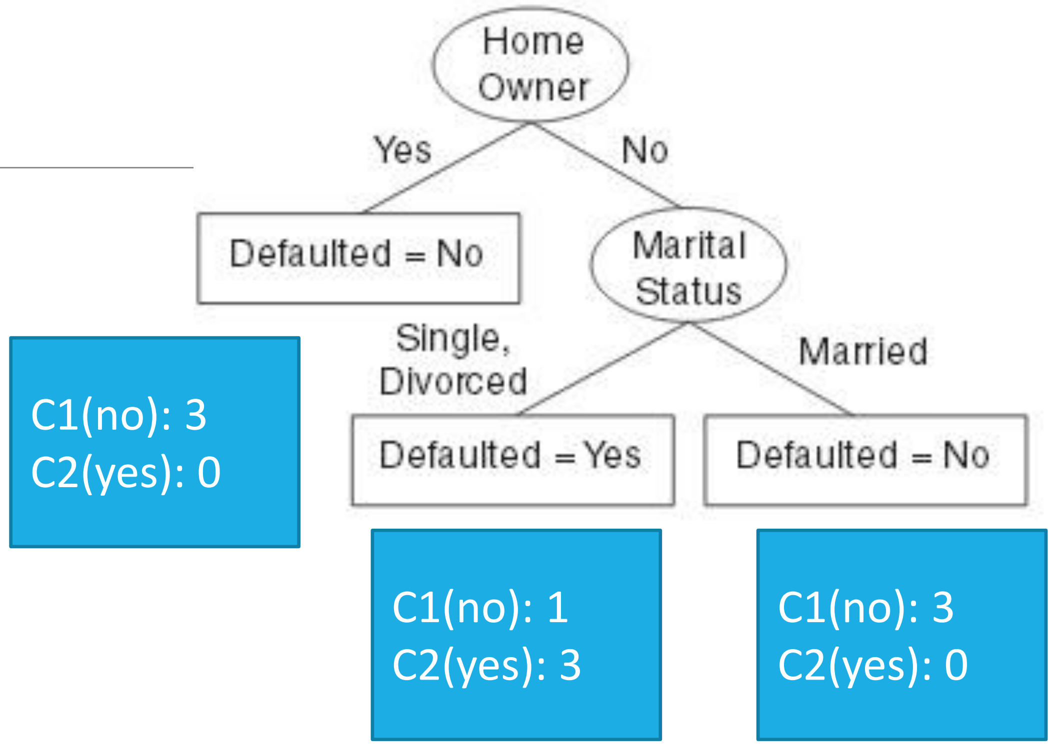
GINI = 1 – (4/7)^2 - (3/7)^2 = .42

Entropy = -(4/7)log2(4/7) - (3/7)log2(3/7) = .98 (also not pure)

The left node is pure and done.

The right node is not pure and needs to be split again.

Step 3



All "Home Owners" are class: Default=NO. Therefore, that node is pure and does not require further partitioning.

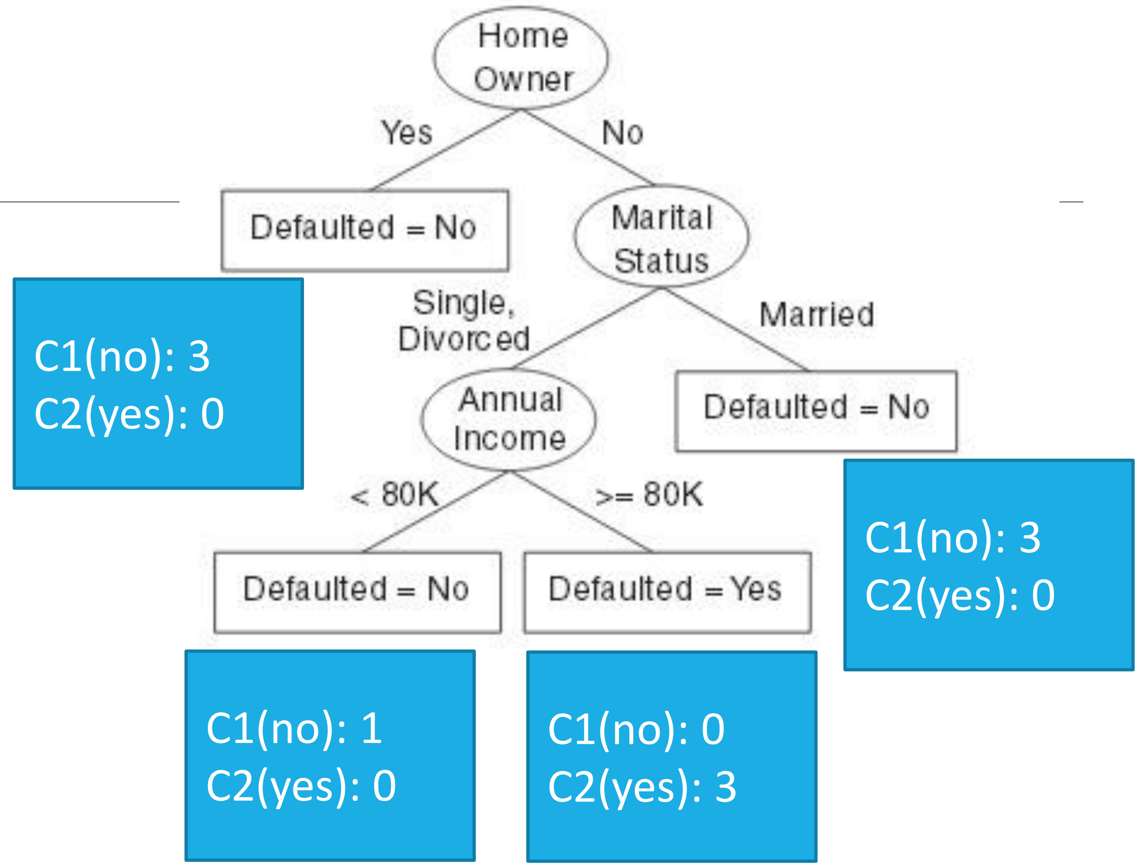
If the borrower is not a home owner, they are further partitioned by Marital Status.

The only node that is still not pure here is "Single/Divorces" AND "not home owner". Another attribute condition must be added.

Step 4

"Annual Income" is used as an attribute condition with <80K, or >80K.

Now, all nodes are pure and the leaf nodes contain the classification.



Test it!
Suppose a non-married person with 75K per year who does not own a home gets a loan – **<u>will they pay it back?</u>**

About Hunt

Hunt works well if the training set contains every combination of all possible attributes.

What happens if it does not?

Design Issues with Decision Trees

How should training records be split?

- How can the “best” attribute test condition be selected?

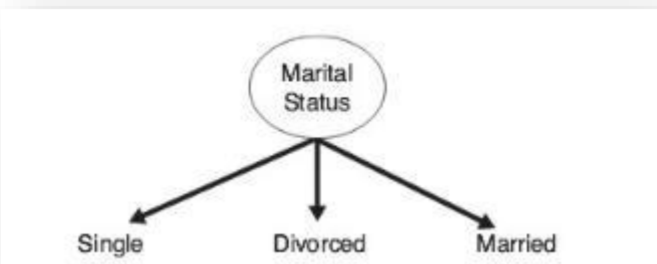
How should the splitting stop?

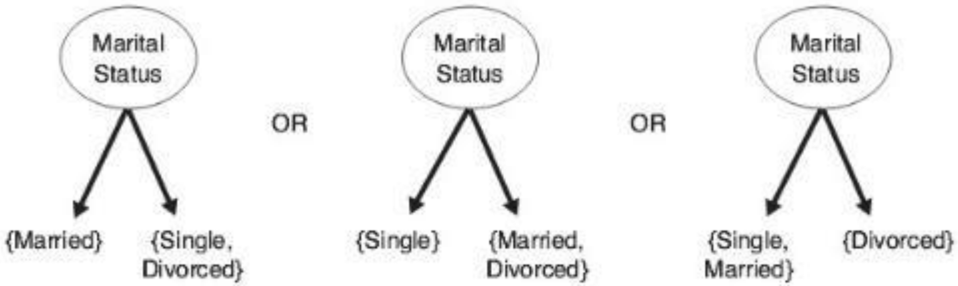
- One option is to expand a node until all records are in the same class or have identical attribute values.

Expressing Attribute Test Conditions:

- **Binary attributes**: two possible outcomes such as married or not married.
- **Nominal Attributes**:
  - Multi-split – one node for each attribute name
  - binary split (CART does this) determined by investigating the best of the $2^k - 1$ options for splitting. (k is the number of attributes)

Multi-split

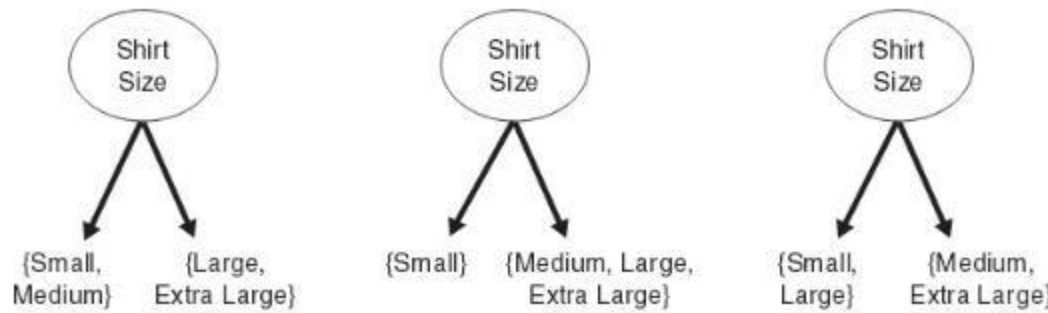




Binary split options for 3 attributes

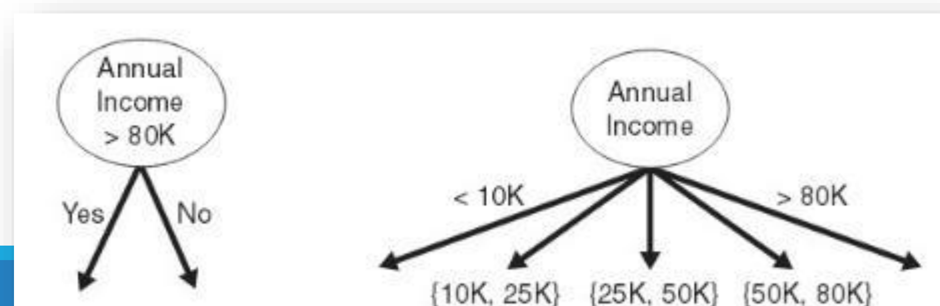
Expressing Attribute Test Conditions:

- **Ordinal attributes**: can also be split using binary or multi.
  - **Why is the last option here not as good?**



Continuous Attributes:

- Can use **comparison set**: (A < x) OR (A >= x)
- Can use a range of options (for mult):

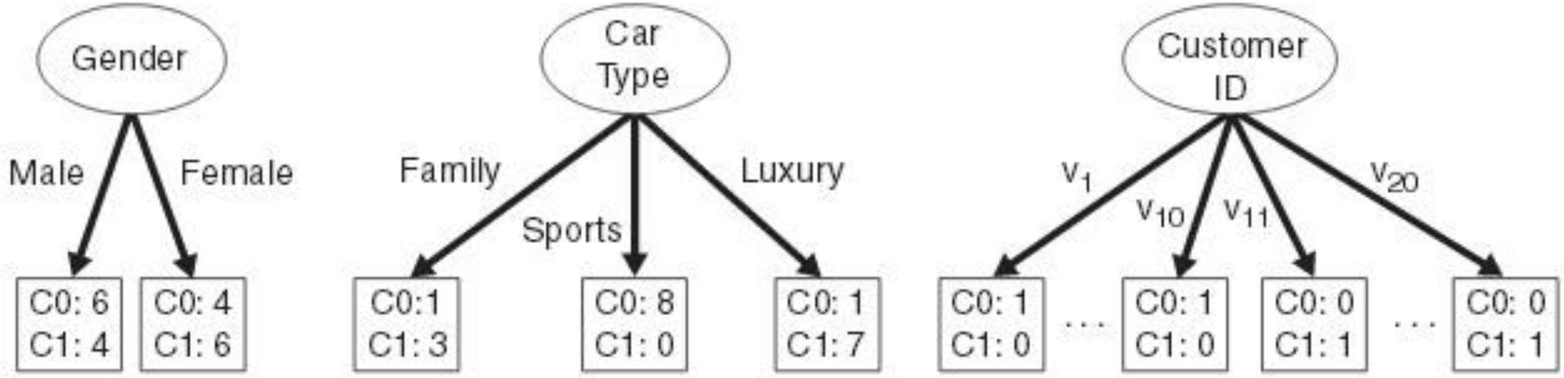


http://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf

Comparing Splits

Note: C0:6 means that there are 6 records of class “0” in the partition.

Which split created purer classes?



Looking at Splits

Car Type
p(C0|Family) = 1/8
P(C1|Family) = 7/8

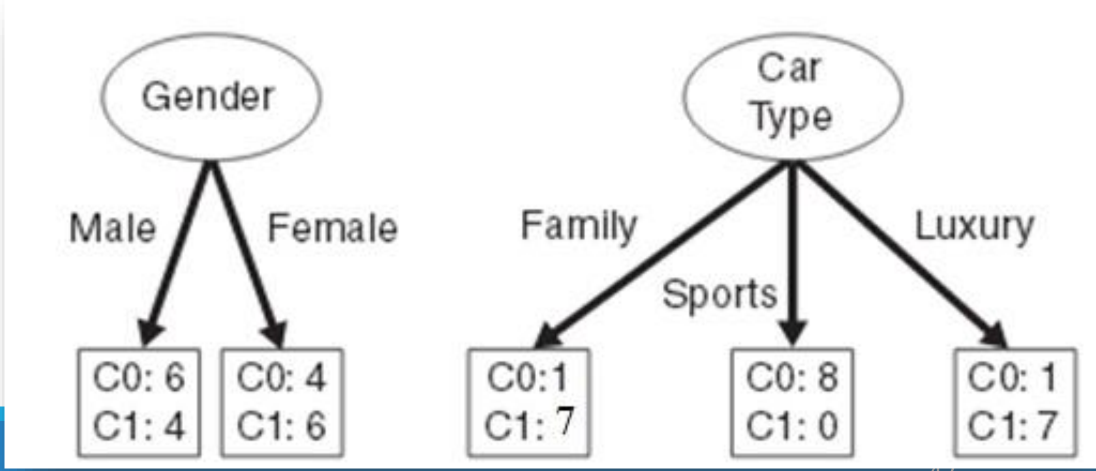
Gender

Left

p(C0|Male) = 6/10
p(C1|Male) = 4/10

Right

p(C0|Female) = 4/10
p(C1|Female) = 6/10



Comparison

Node N_1	Count
Class=0	0
Class=1	6

$$\text{Gini} = 1 - (0/6)^2 - (6/6)^2 = 0$$
$$\text{Entropy} = -(0/6)\log_2(0/6) - (6/6)\log_2(6/6) = 0$$
$$\text{Error} = 1 - \max[0/6, 6/6] = 0$$

Node N_2	Count
Class=0	1
Class=1	5

$$\text{Gini} = 1 - (1/6)^2 - (5/6)^2 = 0.278$$
$$\text{Entropy} = -(1/6)\log_2(1/6) - (5/6)\log_2(5/6) = 0.650$$
$$\text{Error} = 1 - \max[1/6, 5/6] = 0.167$$

Node N_3	Count
Class=0	3
Class=1	3

$$\text{Gini} = 1 - (3/6)^2 - (3/6)^2 = 0.5$$
$$\text{Entropy} = -(3/6)\log_2(3/6) - (3/6)\log_2(3/6) = 1$$
$$\text{Error} = 1 - \max[3/6, 3/6] = 0.5$$

1) which has lowest impurity?
2) Which has highest impurity?

Example: Splits and Information

Label	Gender	Cholesterol	MaritalStatus	Weight	Height	StressLevel
Risk	M	251	S	267	70	5
NoRisk	F	105	M	103	62	1
NoRisk	F	109	M	100	63	2
Risk	M	198	S	210	70	4
Risk	F	189	S	189	64	3
NoRisk	F	121	S	105	65	1
Risk	M	250	S	156	69	5
NoRisk	M	118	M	190	71	3
Risk	F	290	M	300	62	4
NoRisk	F	156	M	119	69	1

What is the Label?

What options are there for the root node?

Suppose we look at StressLevel for the root. How many SPLIT options do we have?

Example: Splits and Information

Label	Gender	Cholesterol	MaritalStatus	Weight	Height	StressLevel
Risk	M	251	S	267	70	5
NoRisk	F	105	M	103	62	1
NoRisk	F	109	M	100	63	2
Risk	M	198	S	210	70	4
Risk	F	189	S	189	64	3
NoRisk	F	121	S	105	65	1
Risk	M	250	S	156	69	5
NoRisk	M	118	M	190	71	3
Risk	F	290	M	300	62	4
NoRisk	F	156	M	119	69	1

Options to split StressLevel:

1) 5- way split choice

2) 4-way split choices – such as: **4&5** and **1**, **2**, and **3**

3) 3 – way split choices – such as **5** and **3&4** and **1&2**

4) 2 – way split choices – such as **4&5** and **1&2&3**

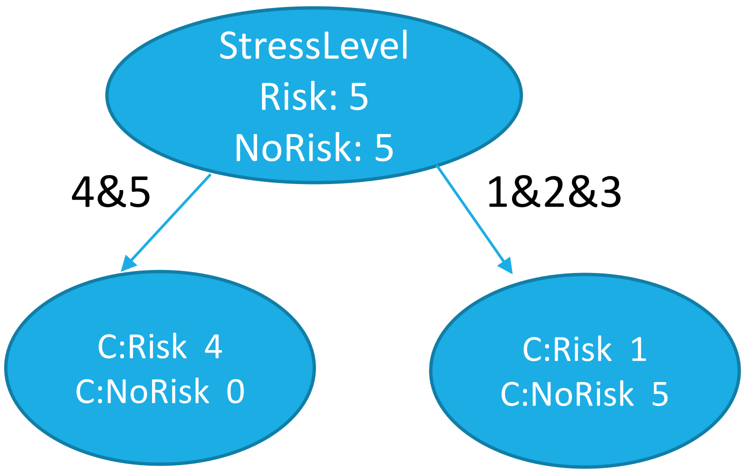
Which creates a better separation of the labels – more information?

Let's compare two of the choices….

Example: Splits and Information

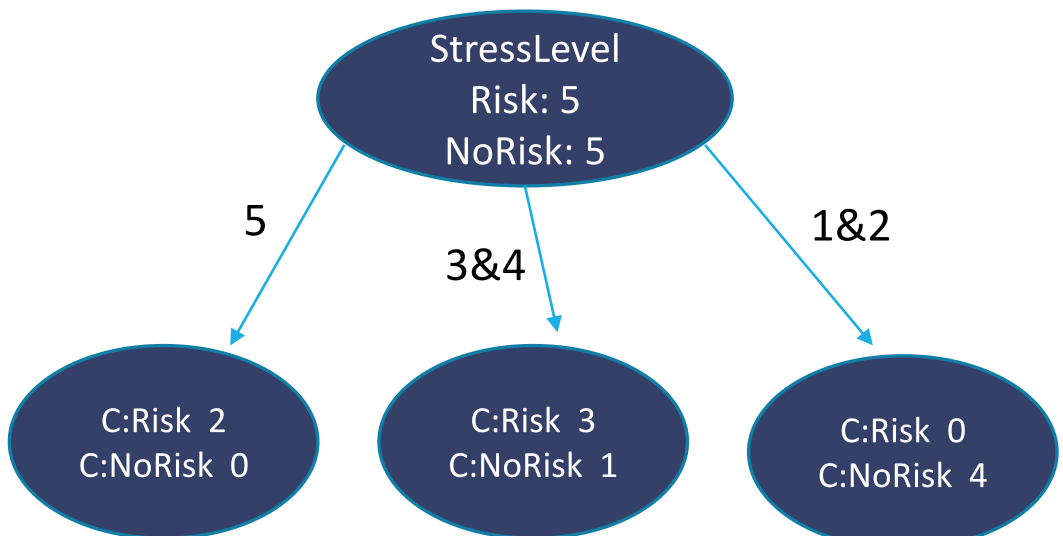
GINI:1- (5/10)^2 –(5/10)^2 = .5 (worst)

Label	Gender	Cholesterol	MaritalStatus	Weight	Height	StressLevel
Risk	M	251	S	267	70	5
NoRisk	F	105	M	103	62	1
NoRisk	F	109	M	100	63	2
Risk	M	198	S	210	70	4
Risk	F	189	S	189	64	3
NoRisk	F	121	S	105	65	1
Risk	M	250	S	156	69	5
NoRisk	M	118	M	190	71	3
Risk	F	290	M	300	62	4
NoRisk	F	156	M	119	69	1



GINI:
1- (4/4)^2 –(0/4)^2
= 0   (pure)

GINI:
1- (1/6)^2 –(5/6)^2
= .278



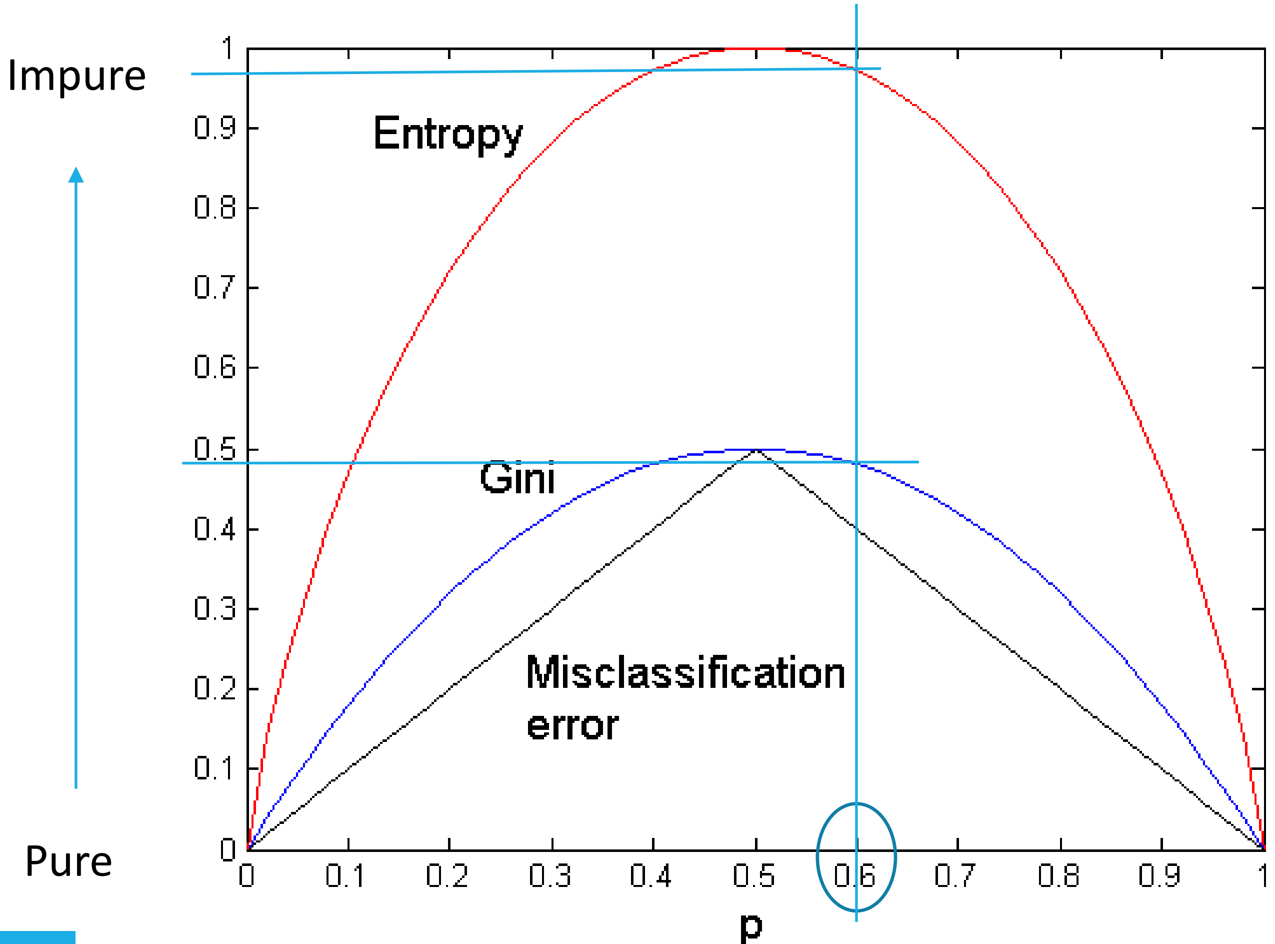
GINI:
1- (2/2)^2 –(0/2)^2
= 0

GINI:
1- (3/4)^2 –(1/4)^2
=.375

GINI:
1- (0/4)^2 –(4/4)^2
=0

Comparison among Splitting Criteria

For a 2-class problem:



Example1:
You have **4 bananas** and **0 apples**.
p=P(bananas)=4/4=1
GINI=0 pure
Entropy=0 pure

Example 2:
You have **3 bananas** and **2 apples**.
p=P(bananas)=3/5=.6
GINI=~.48…
Entropy=~.98…

Information Gain

To determine the **strength of a partition** – compare purity of parent node (before split) to child nodes (after split).

The greater the difference – the better the partition condition.

The **Gain** (Δ) is a measure for **goodness of split**.

I is the **impurity measure** of a node (such as GINI or Entropy)

N is the number of records/vectors/rows at parent node

k is the number of attribute values (variable options)

N(vj) is the number of records in child vj.

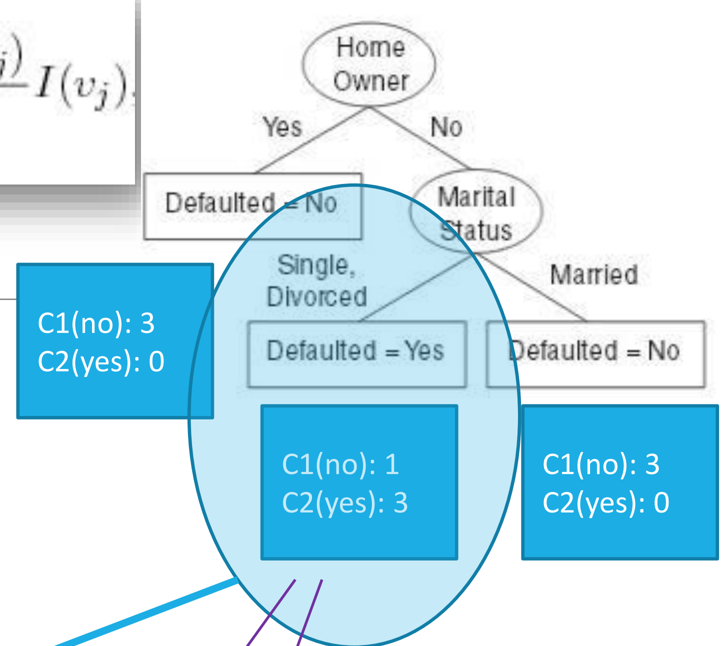
If entropy is used as the impurity measure, the **difference in entropy is the information gain.**

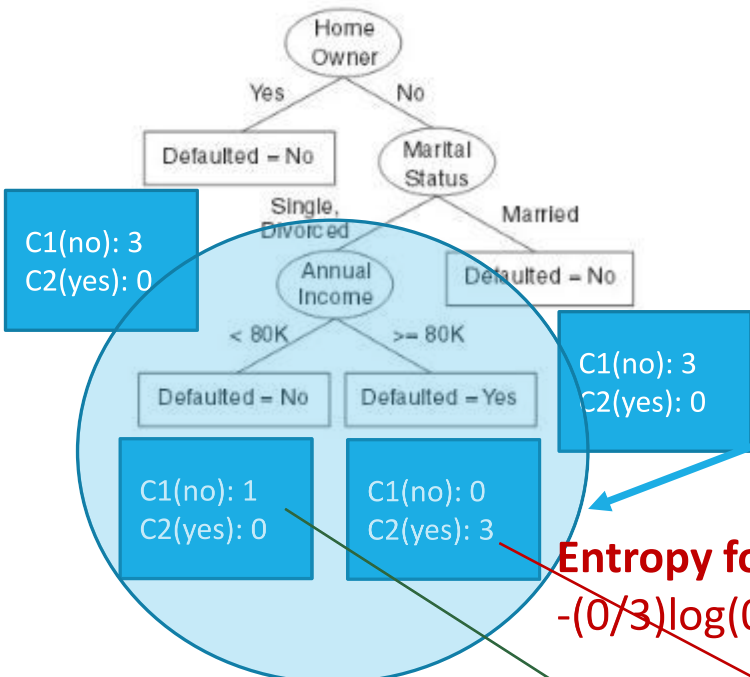
This method is used in **ID3**

$$\Delta = I(\text{parent}) - \sum_{j=1}^{k} \frac{N(v_j)}{N} I(v_j),$$

Example: Information Gain using Entropy as the purity measure. Is information gained?

$$\Delta = I(\text{parent}) - \sum_{j=1}^{k} \frac{N(v_j)}{N} I(v_j)$$





Entropy for Parent =
-(1/4)log(1/4) - (3/4)log(3/4)
=.604

Entropy for right node =
-(0/3)log(0/3) - (3/3)log(3/3)=0

Entropy for left node =
-(1/1)log(1/1) - (0/1)log(0/1) = 0

Information GAIN:
I(Parent) - sum over all children N(v)/N * I(v) =
.604 - [(1)/(4) * 0 + (3)/(4)*0] = .604

The greater the difference between I(Parent) and children – the better the partition condition.

Calculations for Information Gain Using Entropy

Entropy for Parent =

-(1/4)log(1/4) - (3/4)log(3/4) =

-(1/4)(-2) - (1/4)(-.415)= .604

Entropy for left node =

-(1/1)log(1/1) - (0/1)log(0/1) =

0 - 0 = 0

Entropy for right node =

-(0/3)log(0/3) - (3/3)log(3/3) =

0 - 0 = 0

Information GAIN:
I(Parent) - sum over all children N(v)/N * I(v)
=
.604 - (1)/(4) * 0 - (3)/(4)*0 = .604

This is the max possible difference and so is the best partition.

N is the num records at parent
k is the num attribute values (ours has two possible values)

N(v) is the num of records in child
I in this case is the entropy

Decision Tree Based Classification

Advantages:

- Inexpensive to construct
- Extremely fast at classifying unknown records
- Easy to interpret for small-sized trees
- Robust for missing values
- Redundant attributes do not adversely affect accuracy of prediction
- Accuracy is comparable to other classification techniques for many simple data sets

Decision tree issues

Choosing Splitting Attributes

Ordering of Splitting Attributes

Tree Structure

Stopping Criteria

Training Data

Pruning

Occam's Razor

Given two models of similar generalization errors, one should **prefer the simpler model over the more complex model**

For complex models, there is a greater chance that it was fitted accidentally by errors in data

Therefore, one should include model complexity when evaluating a model