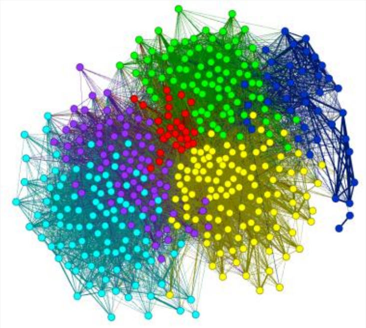
Data Analytics

Lecture 6



Outline

- Modeling and Machine Learning part 1
- Decision Trees
- Instance Based Learning

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

Adapted from
Introduction to Data Mining
Tan, Steibeck, Kumar

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.

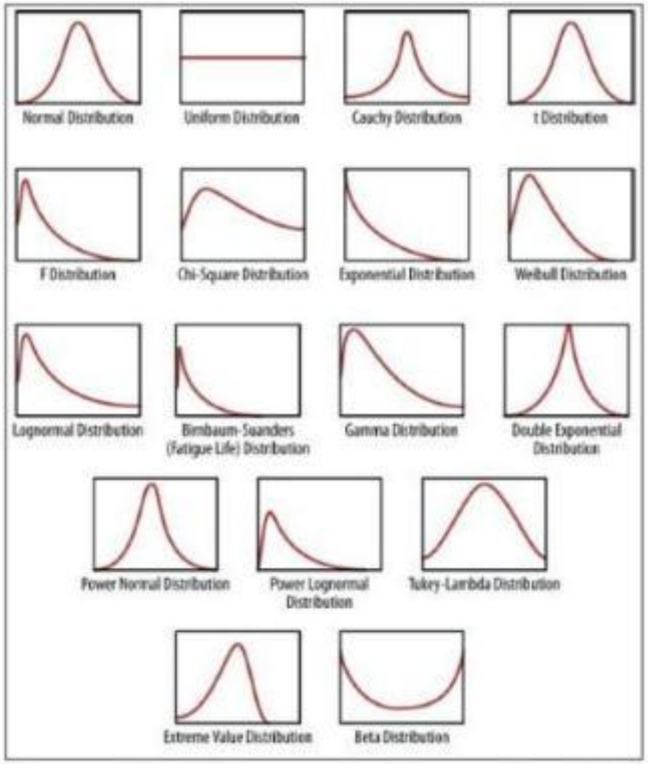
Statistical Modeling

- A way to express a **model using mathematics**
- The model designer makes an assumption about the **generative process** of the data
- 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

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

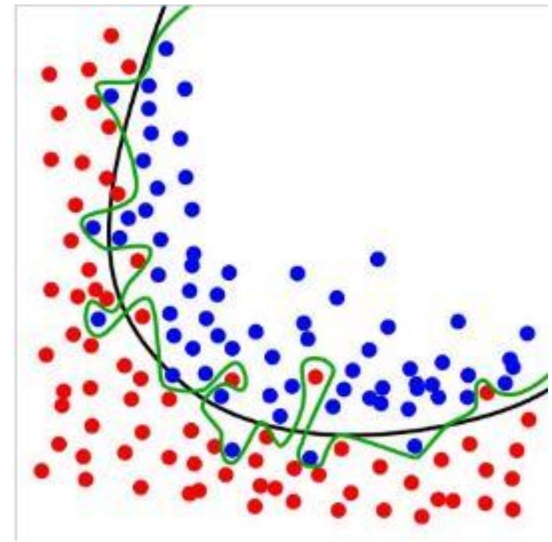
Deciding on a model to use

- Conduct **exploratory analysis**
- Develop a **hypothesis** to test
  - **Try a linear function first – why?**
    - Write down assumptions
    - Does this make sense?
  - If necessary, begin looking at more sophisticated models
    - Write assumptions
    - Does this make sense?

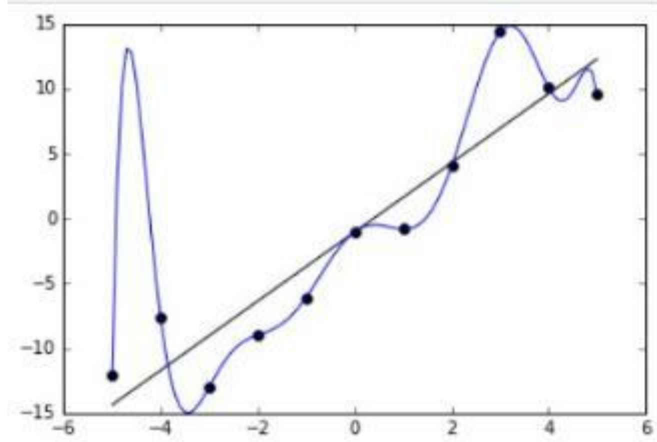
Fitting a Model

- 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

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

- 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 sets used to build the model and the test set used to validate it.
- Training sets and testing sets should not overlap in values.
- **Cross-validation** (leave-one-out) is often used.

Concepts for ML Classification

Input data: collection of records (also called an instance or example).

- For example: tuple(**x**, *y*), where **x** is the set (vector) of known attributes (variable values) and *y* is the **class label (called the target)**.

Classification: learning a target/class **function** f that maps any vector of attributes, **x** to a predefined class y.

f: **x** → y   f is a classification model

- **Descriptive Modeling**: Classification model that can distinguish between objects of different classes.
- **Predictive Modeling**: Using a classification model to predict a label/class given a vector/record **x**

Example: Feature Table

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

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

1. **What are the classes, y?**
2. **What are the records, x?**

Answer:

- The first record $\mathbf{x}_1$ is:

(human, warm-blooded, hair, yes, no, no, yes, no)
The label or class y is mammal

- The second record $\mathbf{x}_2$ is:

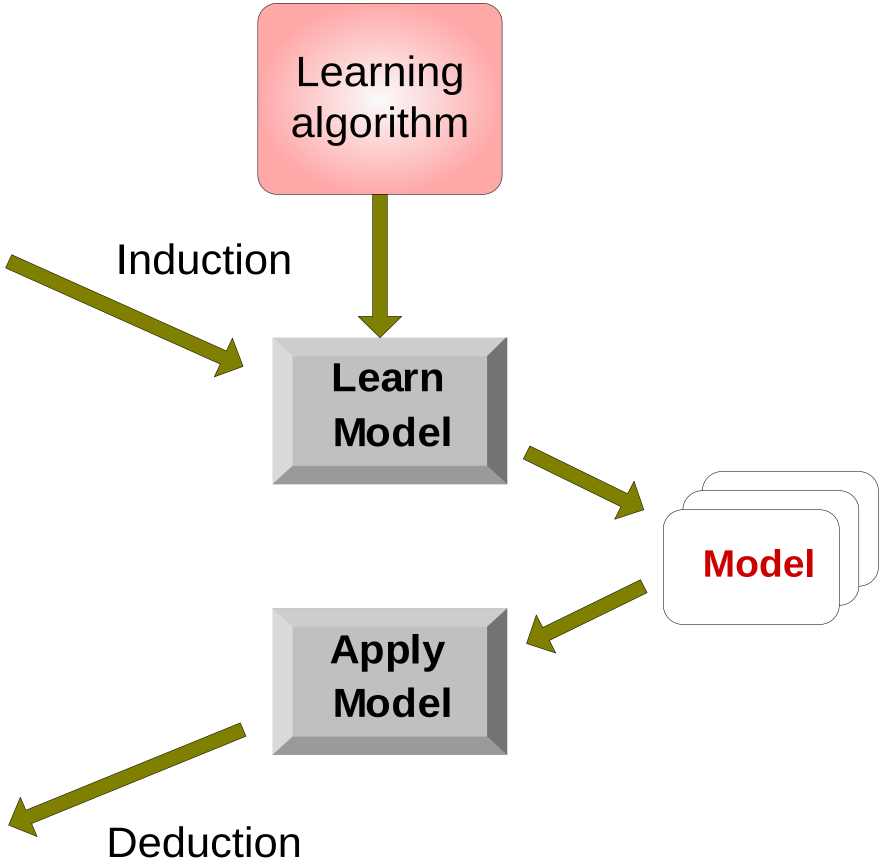
(python, cold-blooded, scales, no, no, no, no, yes)
The label or class y is reptile.

Given a new vector x_n
(grib, warm-blooded, hair, yes, no, no, yes, no)
Predict the class?

Illustrating A 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



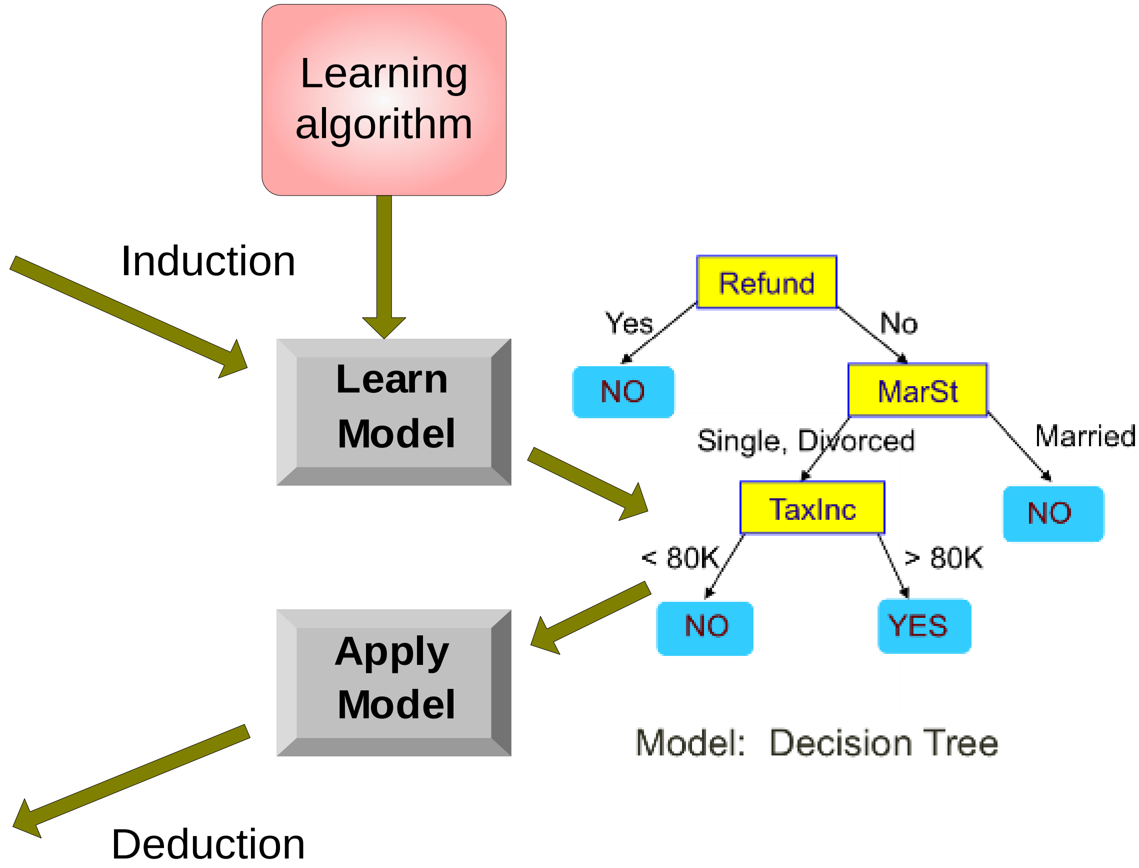
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

EXAMPLE: Is Someone Cheating on Their Taxes?

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	?

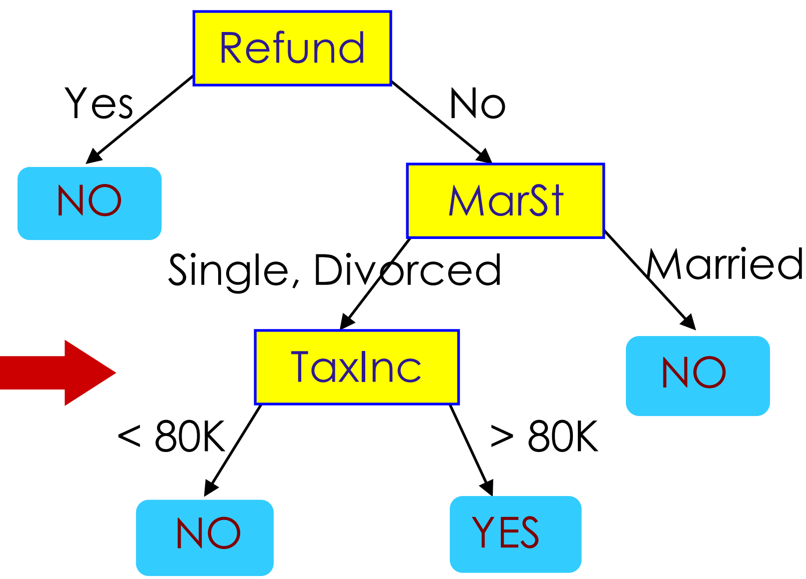
Test Set

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

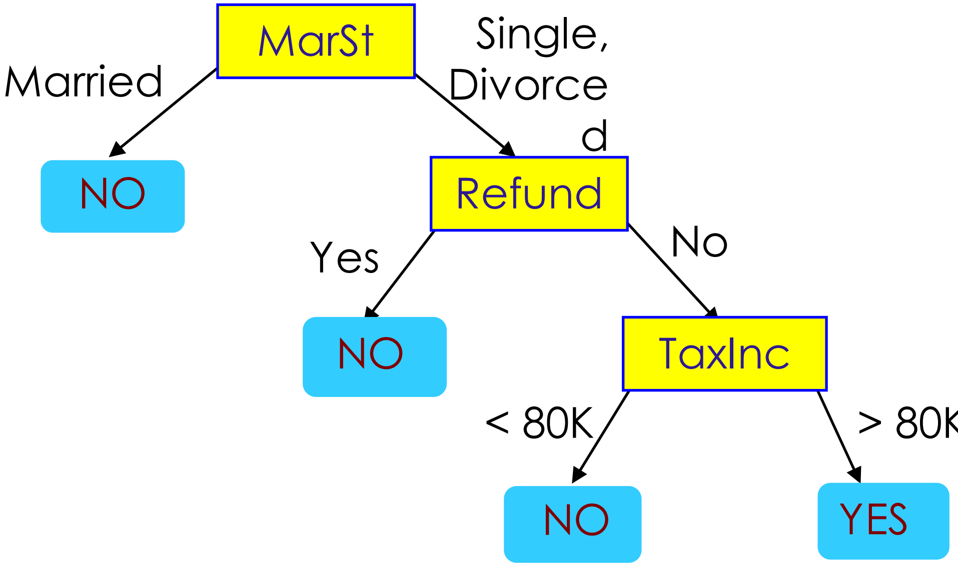


Model: Decision Tree

Another Example of Decision Tree – there are infinite tree options

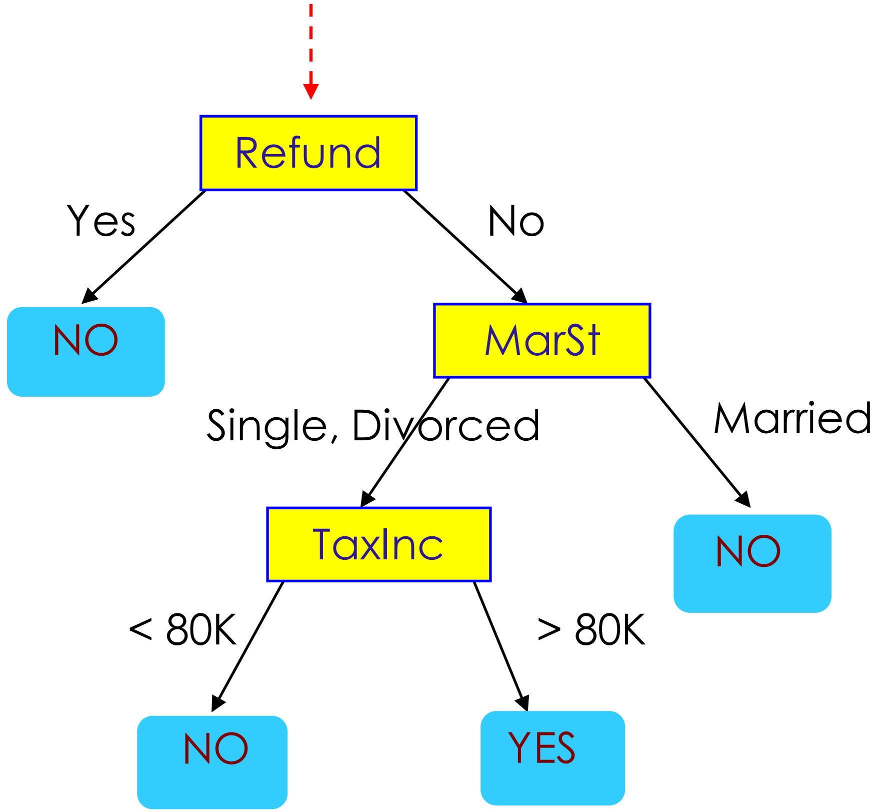
categorical categorical continuous class

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**



Apply Model to Test Data

Start from the root of tree.



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Performance Evaluation

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

Is accuracy always a good measure?

Can you think of an example when it is not?

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

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

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.

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**

Cost vs Accuracy

Count	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a	b
	Class=No	c	d

Cost	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	**p**	**q**
	Class=No	**q**	**p**

Accuracy is proportional to cost if

Proof:

1. C(Yes|No)=C(No|Yes) = q
2. C(Yes|Yes)=C(No|No) = p

$N = a + b + c + d$

$\rightarrow b + c = N - a - d$

Accuracy = (a + d)/N

$$\begin{aligned}\text{Cost} &= p\,(a + d) + q\,(b + c) \\ &= p\,(a + d) + q\,(N - a - d) \\ &= p\,(a + d) + qN - q(a + d) \\ &= q\,N - (q - p)(a + d) \\ &= qN - N(q - p)\,\text{Accuracy} \\ &= N\,[q - (q-p)\ \text{Accuracy}]\end{aligned}$$

Classification Techniques

- **Decision Tree Methods**
- Instance-based Methods
- Bayesian algorithms (Naïve Bayes)
- Support Vector Machines
- Ensembles (Random Forest)

Decision Trees

ML Topic 1

Decision Tree Overview

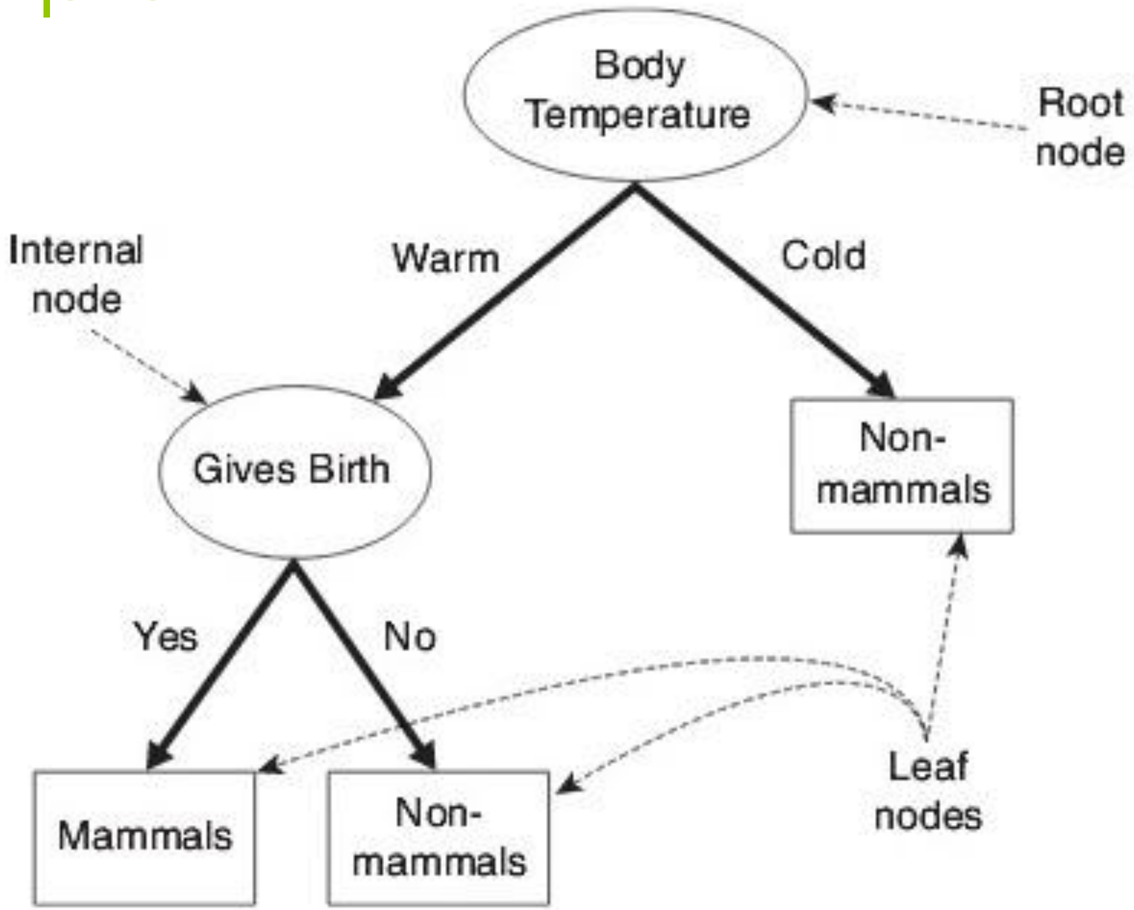
Build a **classifier that is a directional tree structure**.

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 records**, associated with node t in the tree.
- Let **xi** be record i such that yi is the class label.
- All training records are: {**x**1, **x**2, …, **x**n} with associated class labels

{ y1, y2, …, yn}

Method: The tree is created in a **recursive** fashion by continuing to partition the training records (**x**) into purer subsets.

Steps:

1) If all records in D_t belong to the same class y_t, then t is a leaf node labeled as y_t.
2) If D_t contains records that belong to MORE THAN one class, an **attribute test condition** is selected to partition into smaller subsets.
3) The above is recursively repeated

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 record?

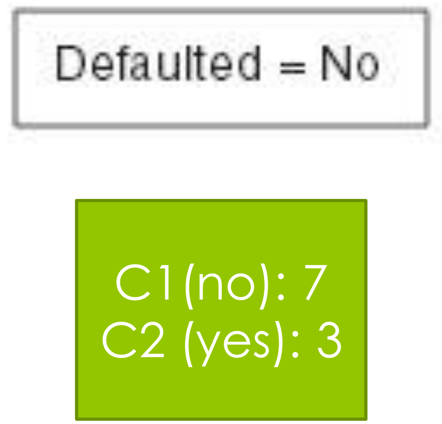
What is the label of each?

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

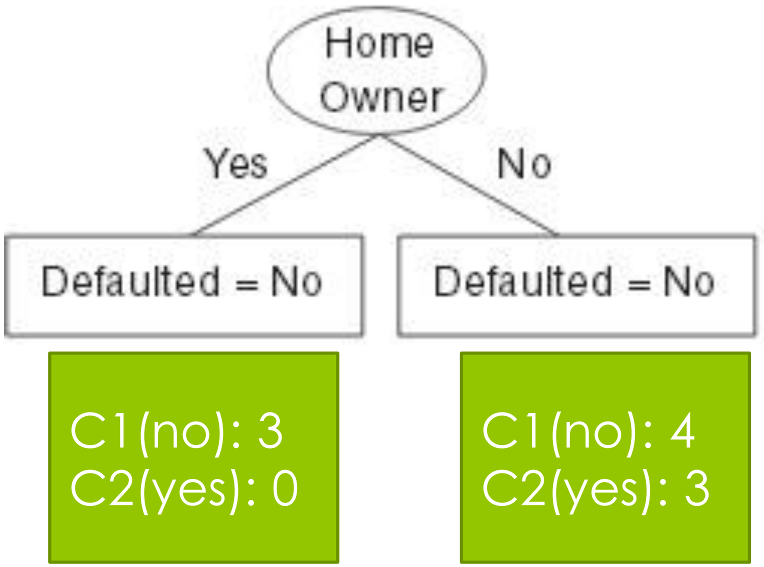
Step 1



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)**.

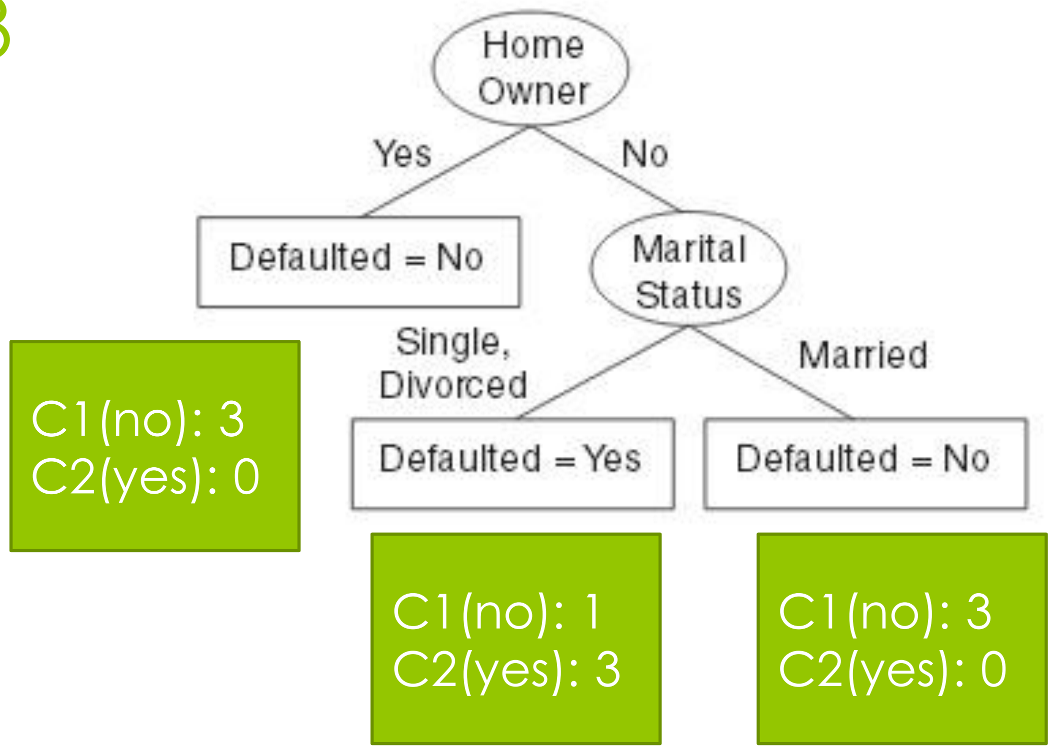
Step 2



Using the attribute condition test of "Home Owner" still results one mixed class, with the majority of each labeled as Default=NO.

This must be further refined until the node is **pure**.

Step 3



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

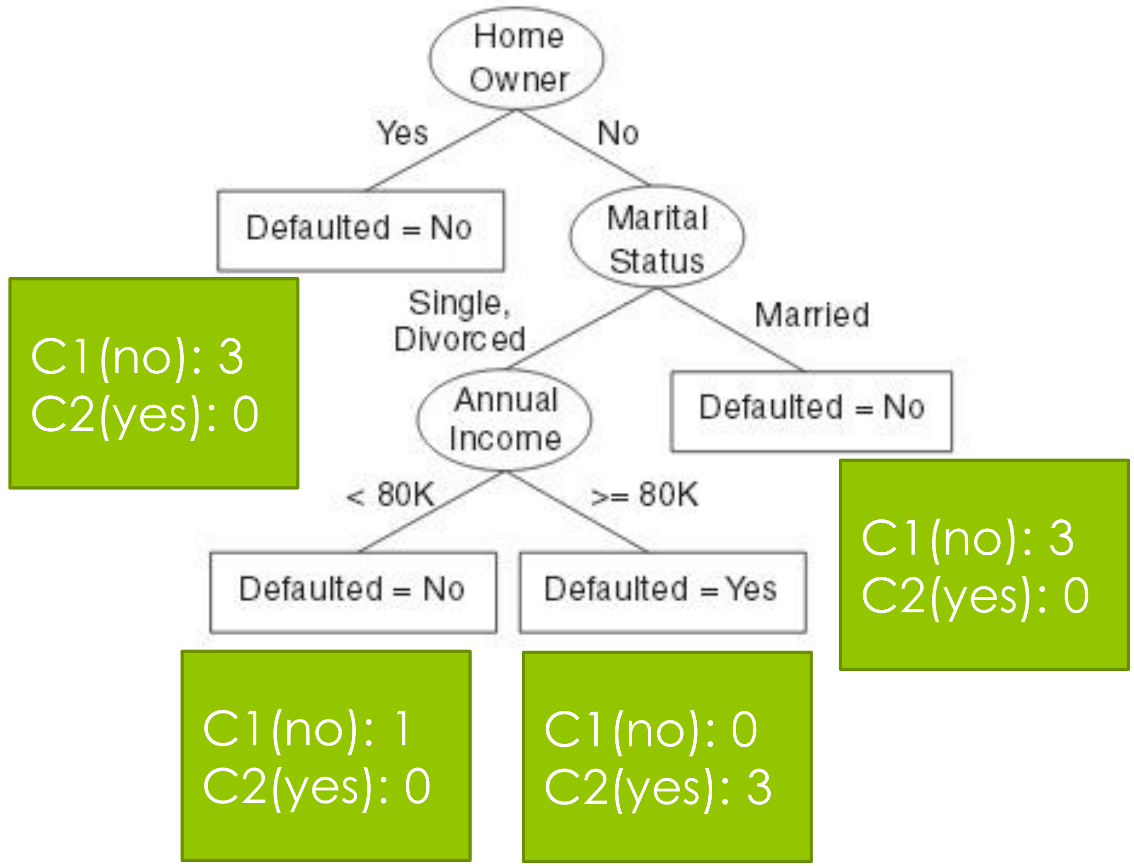
If the borrower is not a home owner, they are further partitioned by Marital Status.

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?

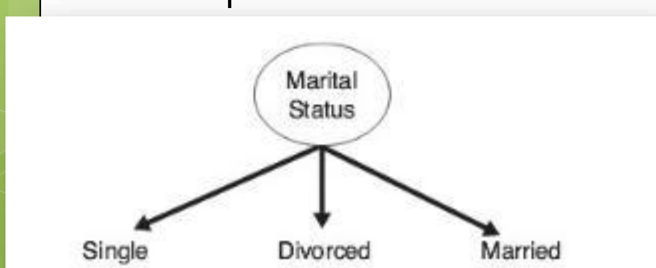
Design Issues with Decision Trees

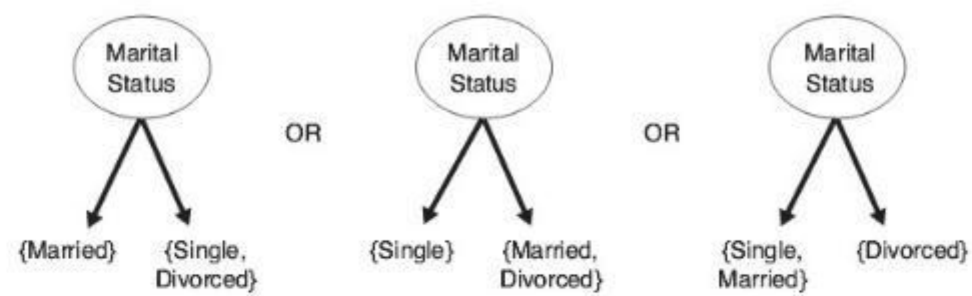
- **How should training records be split**?
  - How can the “best” attribute test condition be selected?
- **How should the splitting stop**?
  - One option is to expand a node until all records are in the same class or have identical attribute values.

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



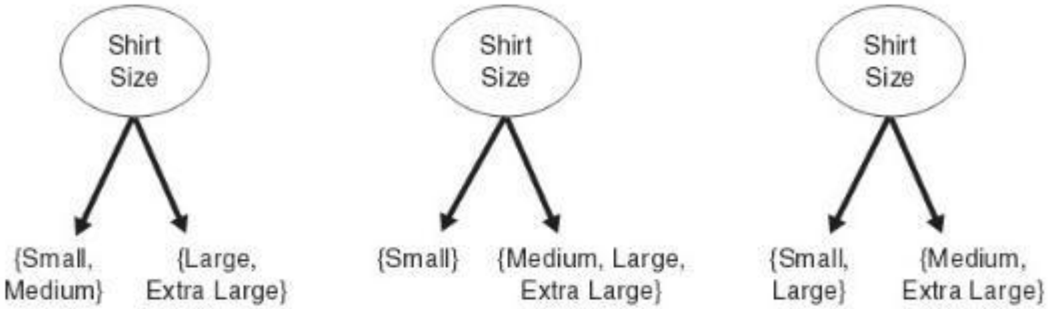


http://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf

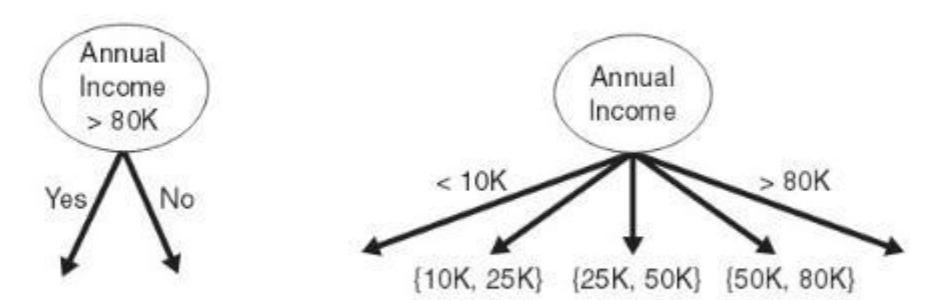
Binary split options for 3 attributes

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):

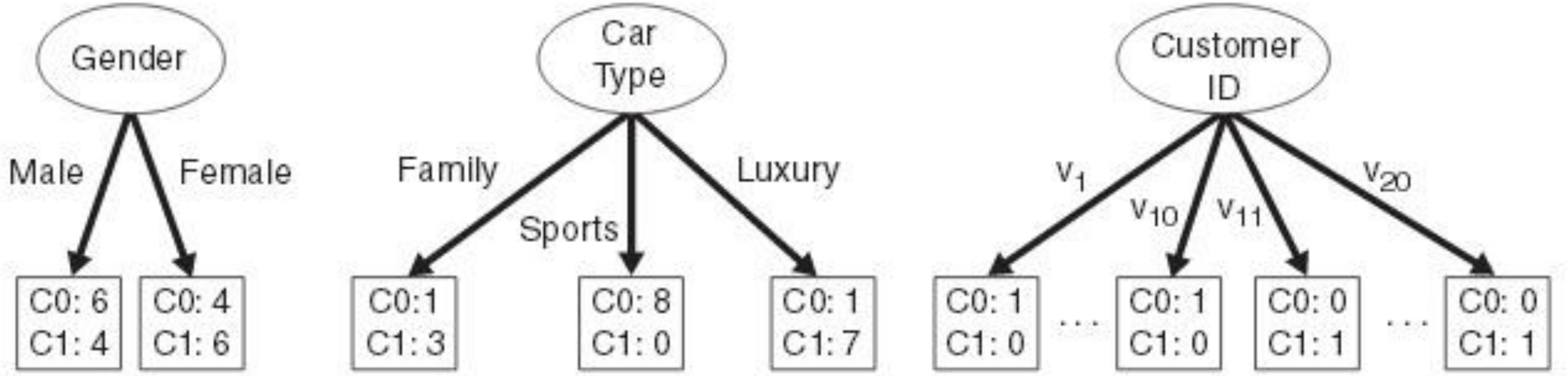


http://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf

Comparing Splits

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

Which split created purer classes?



Methods for Measuring "Best" Split

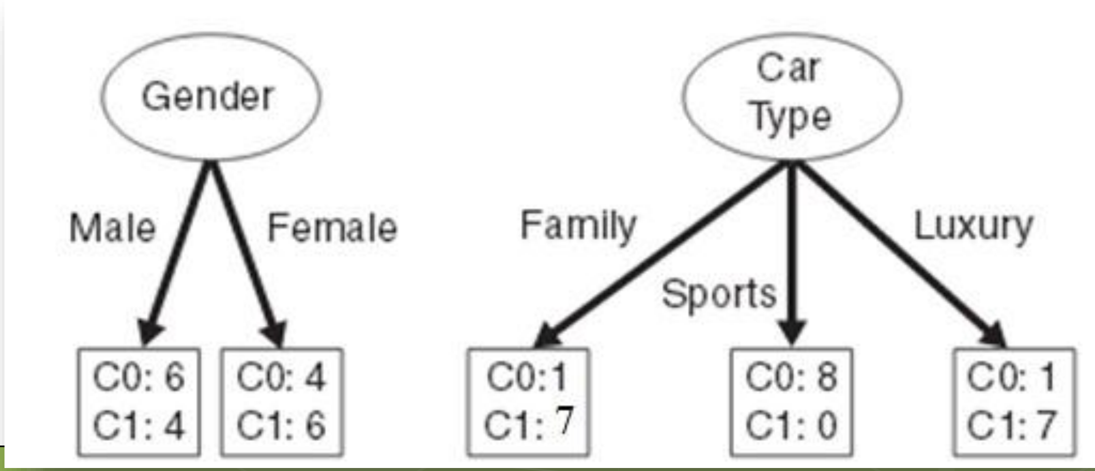
$$\begin{aligned} \text{Entropy}(t) &= -\sum_{i=0}^{c-1} p(i|t)\log_2 p(i|t), \\ \text{Gini}(t) &= 1-\sum_{i=0}^{c-1}[p(i|t)]^2, \\ \text{Classification error}(t) &= 1-\max_i[p(i|t)], \end{aligned}$$

Step 1: Class Node Probability

Let p(i|t) be the fraction of records belonging to class i at a given node t.

For the Gender Partition:
p(C0|Male) = 6/10
p(C1|Female) = 6/10

For the Car Type partition
p(C0|Family) = 1/8



Calculating Entropy

$$\text{Entropy}(t) = -\sum_{i=0}^{c-1} p(i|t) \log_2 p(i|t),$$

Node N_2	Count
Class=0	1
Class=1	5

p(i|t) = p(class 0 | node N2) = 1/6 (recall that i is the class)
p(i|t) = p(class 1 | node N2) = 5/6

Negative sum over all classes:

entropy = -(1/6)log2(1/6) – (5/6)log2(5/6) = .65

Entropy ranges from 0 to 1, where 0 is pure and 1 is the worst case.

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

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

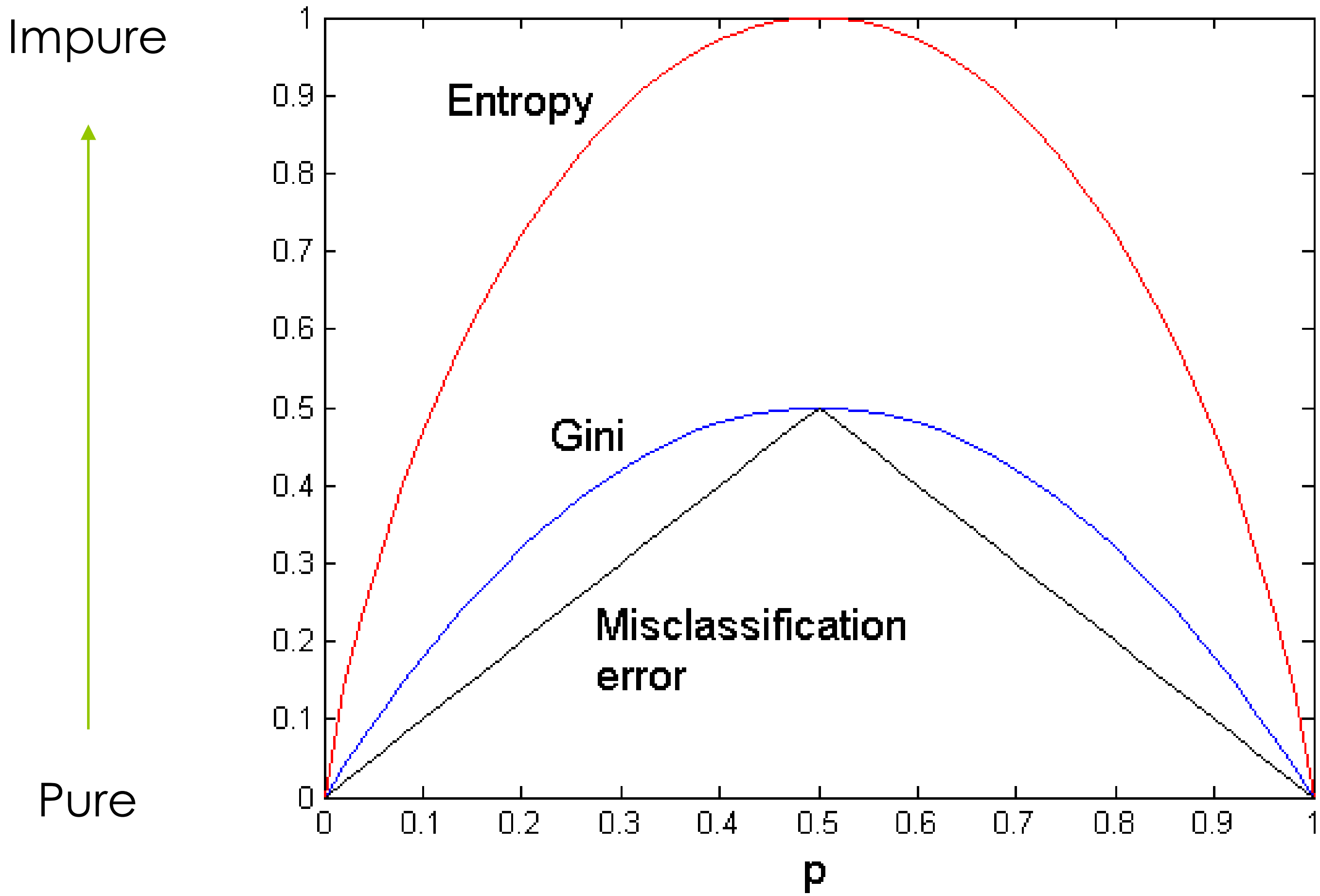
1) which has lowest impurity?
2) Which has highest impurity?

Intuition:

- If a partition results in $p(i \mid t) = .5$, it is very poor. Why?
- If a partition results in $p(i \mid t) = 0$, it is pure. Why?
- The lower the impurity of the partition (so the more pure it is), the more skewed the class distribution. Why?
- A node with class distribution of C1:0, C2:10 is skewed and pure.
- A node distribution with C1:5 and C2:5 is has the highest impurity an no skew.

Comparison among Splitting Criteria

For a 2-class problem:

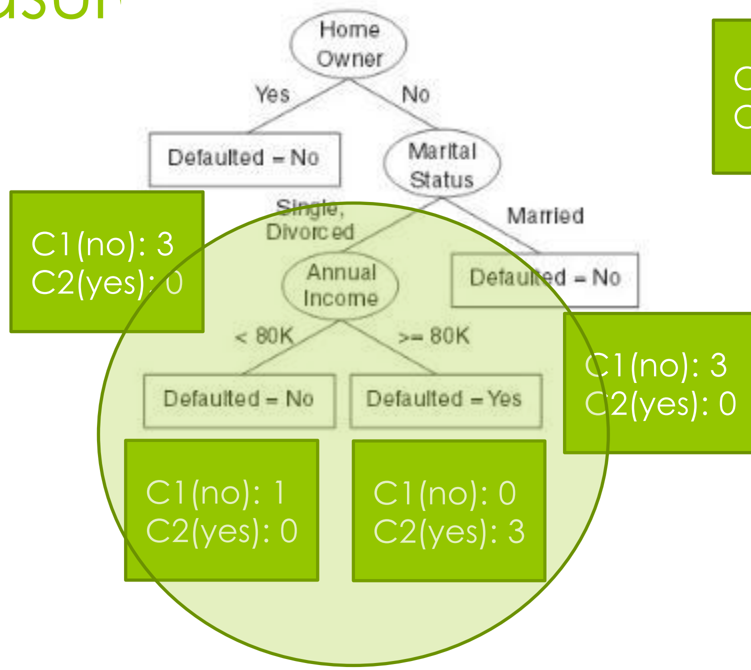


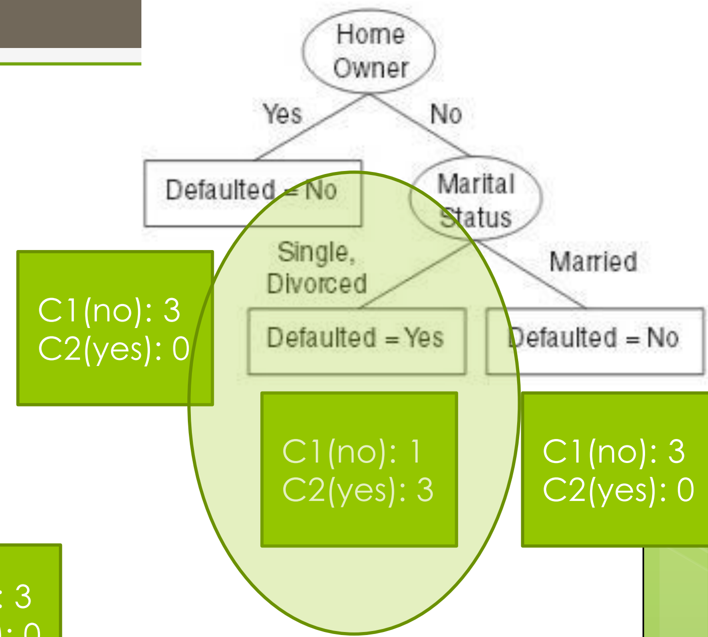
Information Gain

- 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, N is the number of records at parent node, k is the number of attribute values. 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





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

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

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

Splitting Based on Information Gain

- Gain Ratio:

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO}$$

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

Parent Node, p is split into k partitions
n_i is the number of records in partition i

- Used in C4.5

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

Classification Techniques

- Decision Tree Methods
- **Instance-based Methods**
- Bayesian algorithms (Naïve Bayes)
- Support Vector Machines
- Ensembles (Random Forest)

Instance-based Learning

Part 2

Training with Labeled Records

Set of Stored Cases

Atr1	………	AtrN	Class
			A
			B
			B
			C
			A
			C
			B

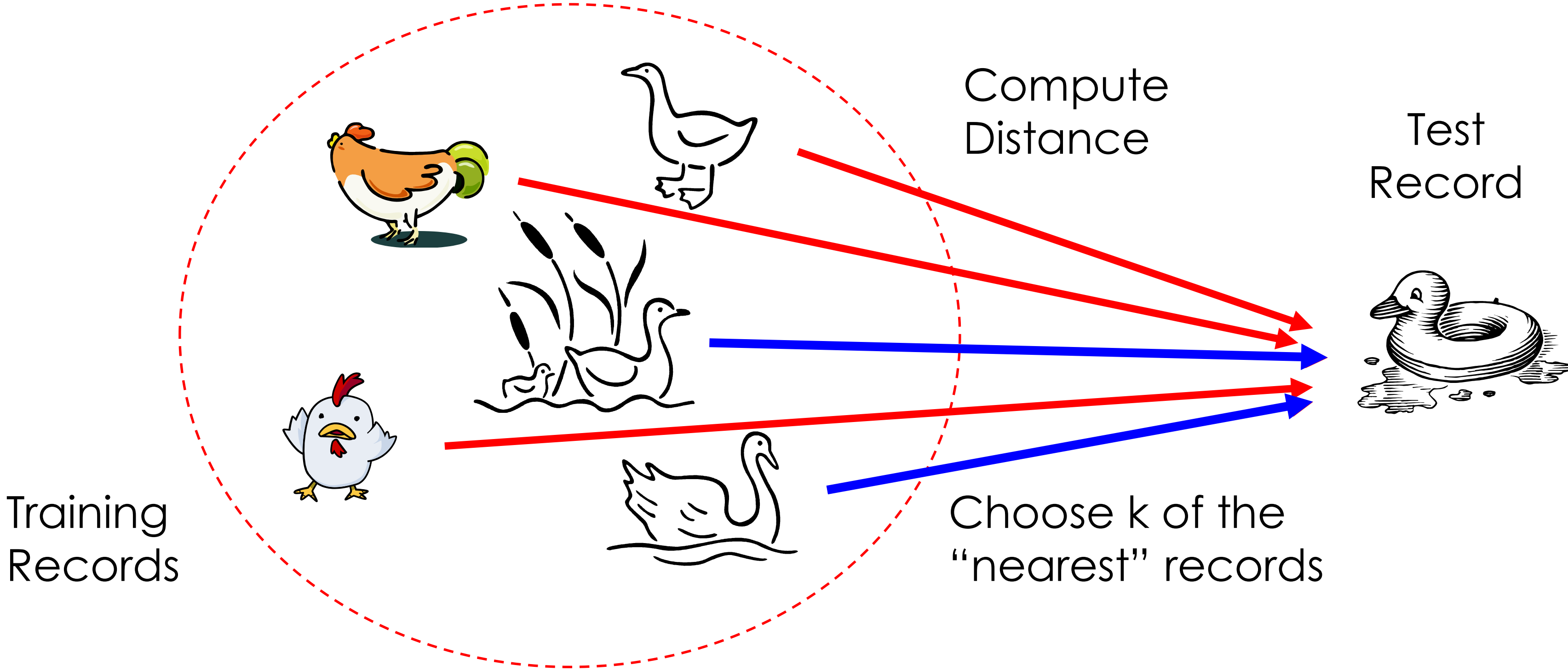
- Store the training records
- Use training records to train a predictor.
- Predict the class label of unseen cases

Instance Based Classifiers

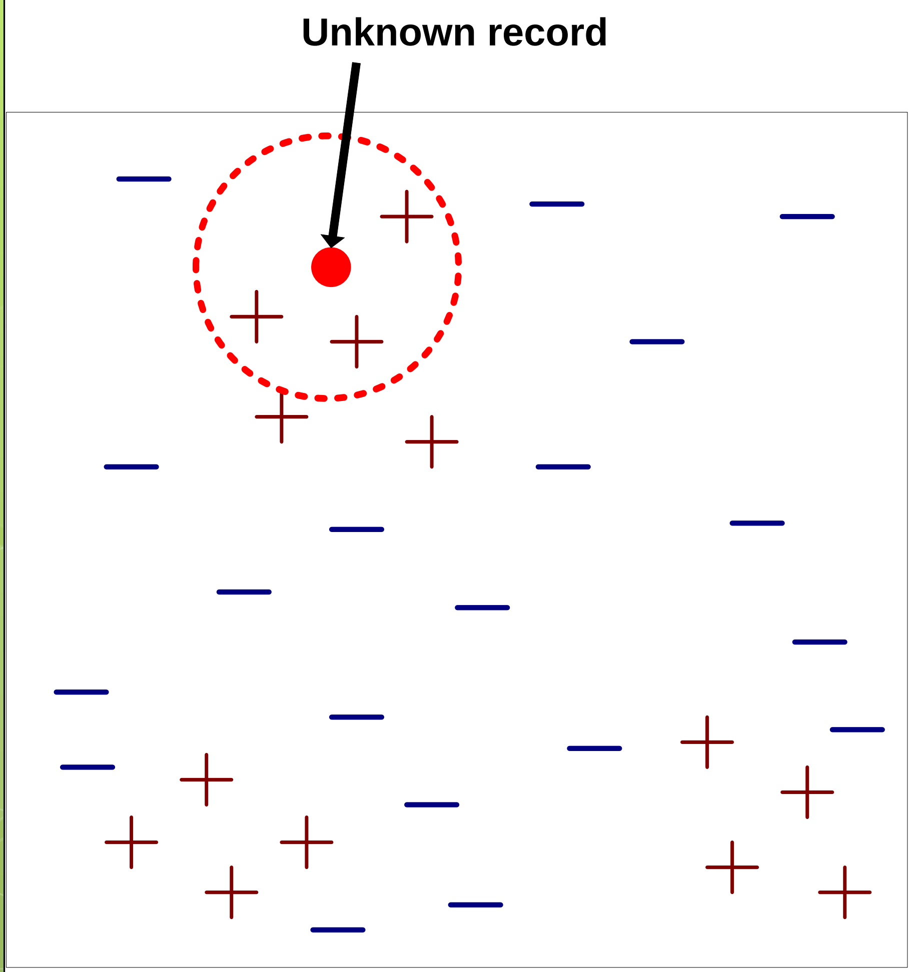
- Examples:
  - **Rote-learner**
    - **Memorizes entire training data** and performs classification only if attributes of record match one of the training examples exactly
  - **Nearest neighbor**
    - Uses k "closest" points (nearest neighbors) for performing classification

Nearest Neighbor Classifiers

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's probably a duck



Nearest-Neighbor Classifiers



- **Requires three things**
  - The set of **stored records**
  - **Distance Metric** to compute distance between records
  - The **value of *k***, the number of nearest neighbors to retrieve
- To **classify an unknown record**:
  - **Compute distance** to other training records
  - **Identify *k* nearest neighbors**
  - **Use class labels of nearest neighbors** to determine the class label of unknown record (e.g., by taking majority vote)

Nearest Neighbor Distance

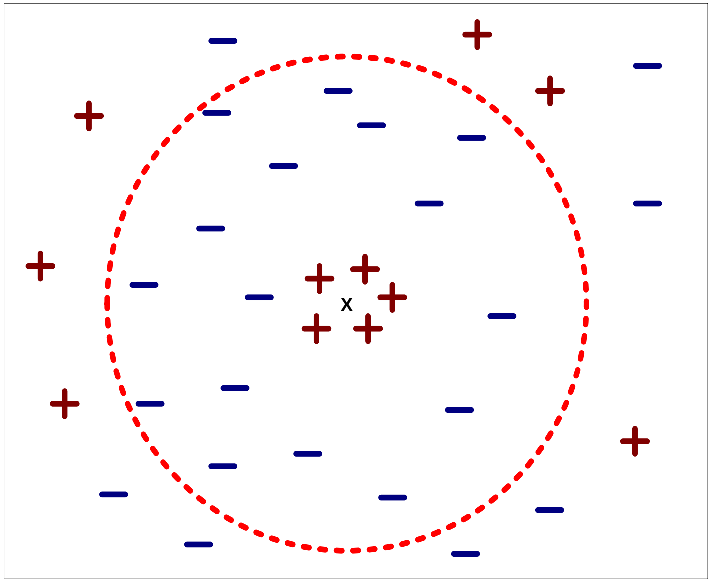
- Compute distance between two points:
  - **Euclidean distance**

$$d(p,q) = \sqrt{\sum_i (p_i - q_i)^2}$$

- Determine the class from nearest neighbor list
  - take the **majority vote** of class labels among the k-nearest neighbors
  - Weigh the vote according to distance
    - weight factor, $w = 1/d^2$

Nearest Neighbor – Determine k

- **Choosing the value of k**:
  - If k is too small, sensitive to noise points
  - If k is too large, neighborhood may include points from other classes



Nearest Neighbor Issues

- **Scaling issues**
  - Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes
  - Example:
    - height of a person may vary from 1.5m to 1.8m
    - weight of a person may vary from 90lb to 300lb
    - income of a person may vary from $10K to $1M

Nearest neighbor summary

- k-NN classifier
  - It does not build models explicitly
  - Unlike eager learners such as decision tree induction and rule-based systems (lazy learner)
  - Classifying unknown records are relatively expensive