CS361
(Software Engineering Program)

Artificial Intelligence II - Applied Machine Learning

Lecture 2
Decision Trees via ID3 – An Application to Business Intelligence *(German Credit Data)* using Weka

Amr S. Ghoneim
(Assistant Professor, Computer Science Dept.)

Helwan University
Fall 2019

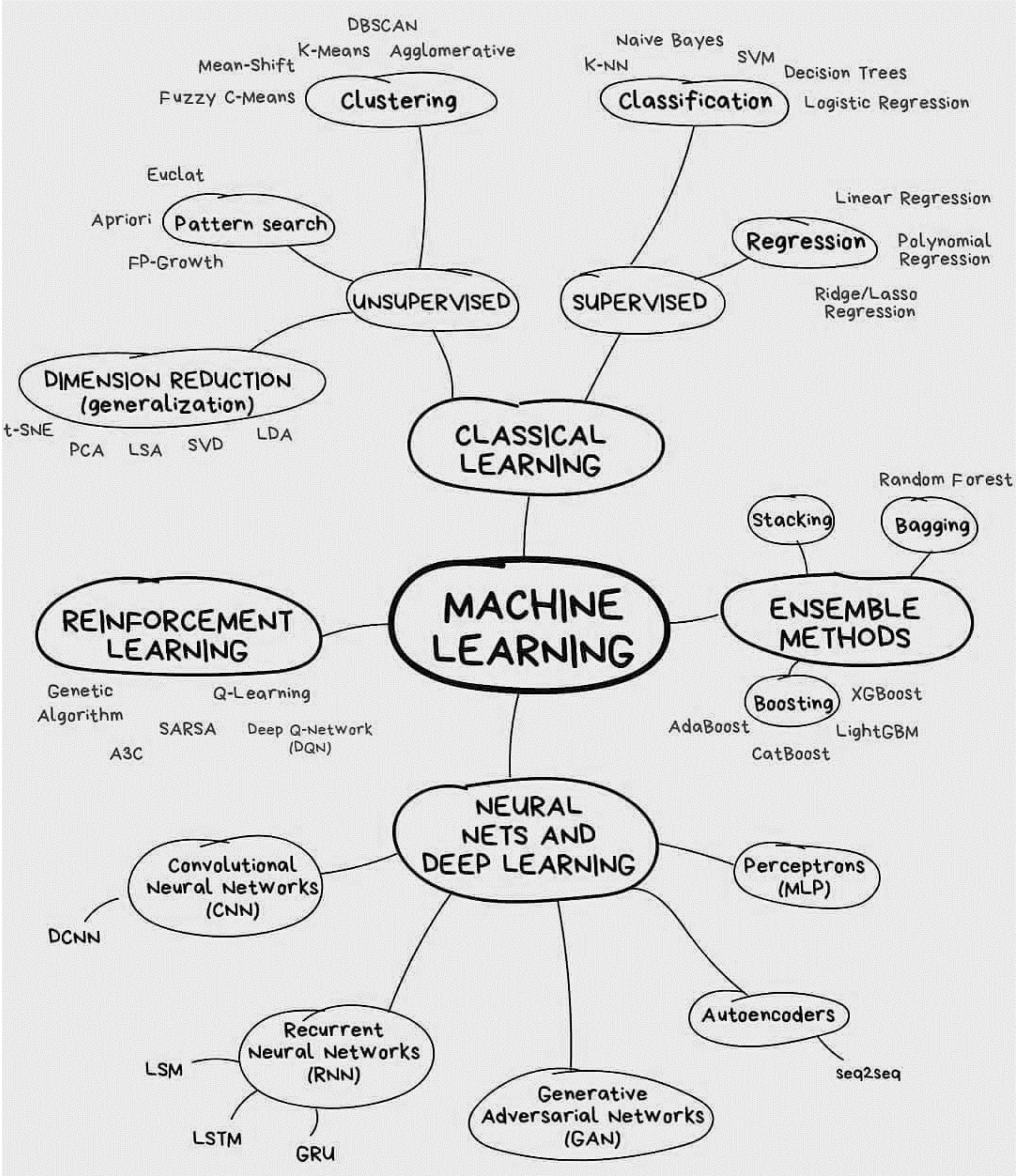
Lecture is based on its counterparts in the following courses:
- *Applied Machine Learning*, **University of Pennsylvania** (School of Engineering and Applied Science)
- *Business Intelligent Systems*, **Monash University** (School of Information Management & Systems)

Today's Key Concepts

- **Machine Learning**
  - **Classical Supervised Machine Learning**
  - **Why use learning?**
  - **Supervised Learning: Training & Testing**
- **Decision Trees *(the ID3 algorithm; a Greedy heuristic – based on information gain – originally developed for discrete features)***
  - **Decision Trees: An Introduction**
  - **Basic Decision Trees Learning Algorithm**
  - **Picking the Root Attribute: Entropy & Information Gain**
  - **Which feature to split on?**
  - **An Illustrative Example**
  - **Decision Trees Induction**
- **Weka; the Waikato Environment for Knowledge Analysis *(Machine Learning Software in Java)***
- **An Application to Business Intelligence**
  - **What is Business Intelligence (BI) & Business Analytics (BA) .. ?**
  - **Why BI? .. Business Management Issues**
  - **A Framework for Business Intelligence (BI's Architecture & Components)**
- **A Case Study: The German Credit Data Analysis (Statlog Dataset)**

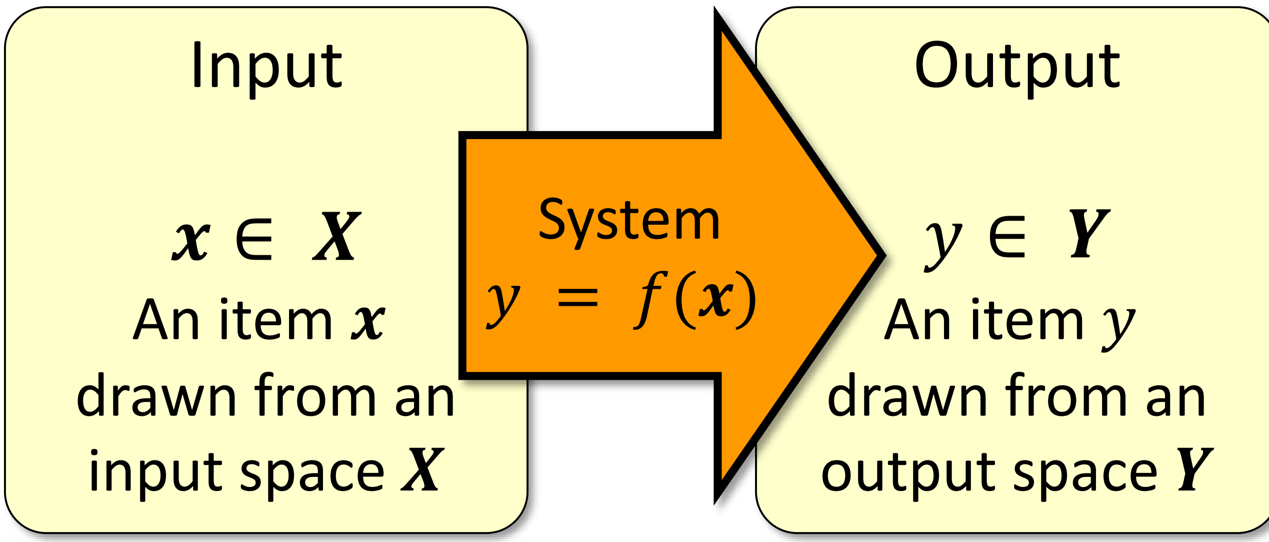
Machine Learning?

{Artificial Intelligence} Machine Learning Map



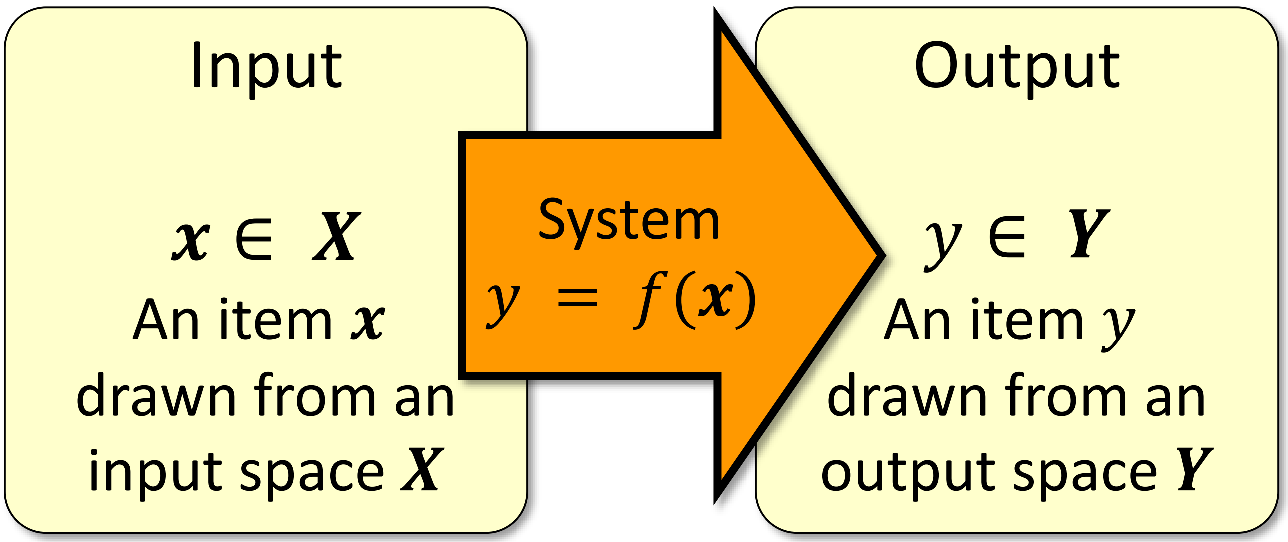
Classical Supervised Machine Learning

Supervised Learning



We consider systems that apply a function $f(\,)$ to input items x and return an output $y = f(x)$.

Supervised Learning

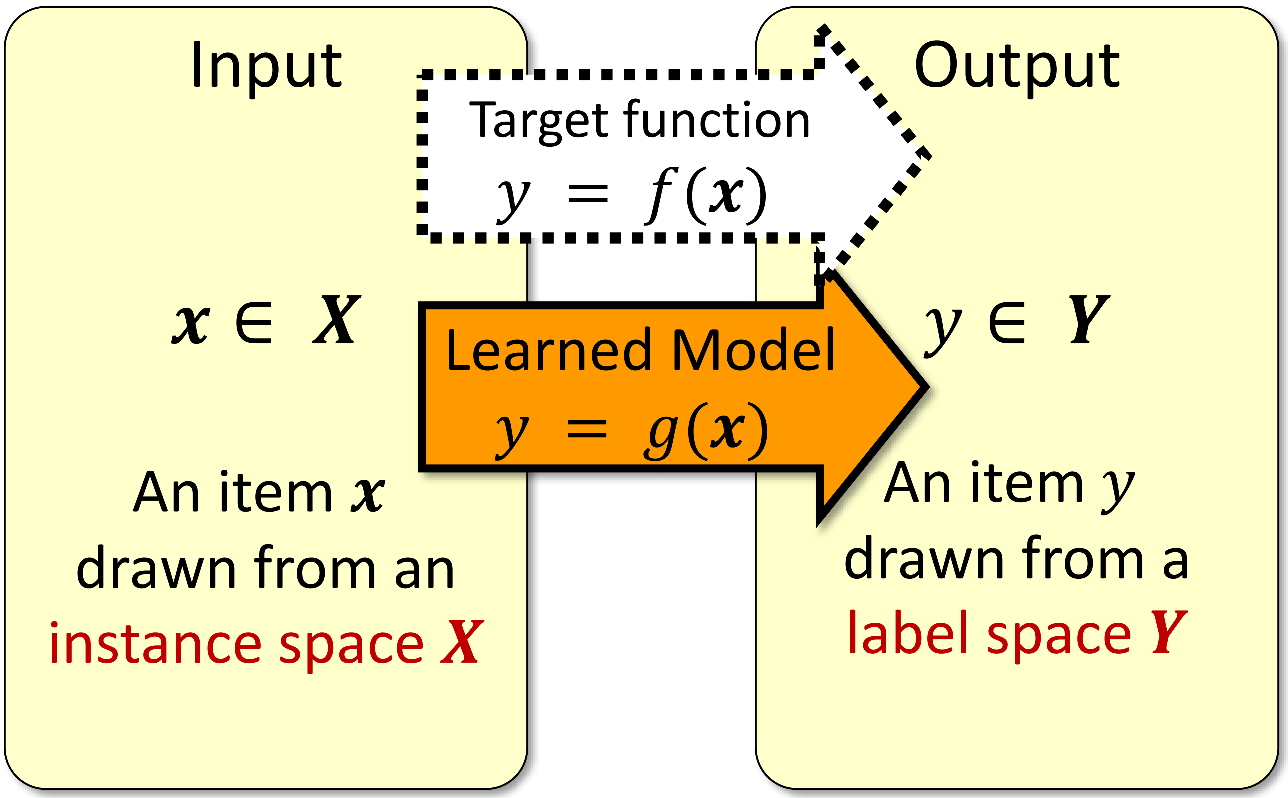


In (supervised) machine learning, we deal with systems whose $f(x)$ is learned from examples.

Why use learning?

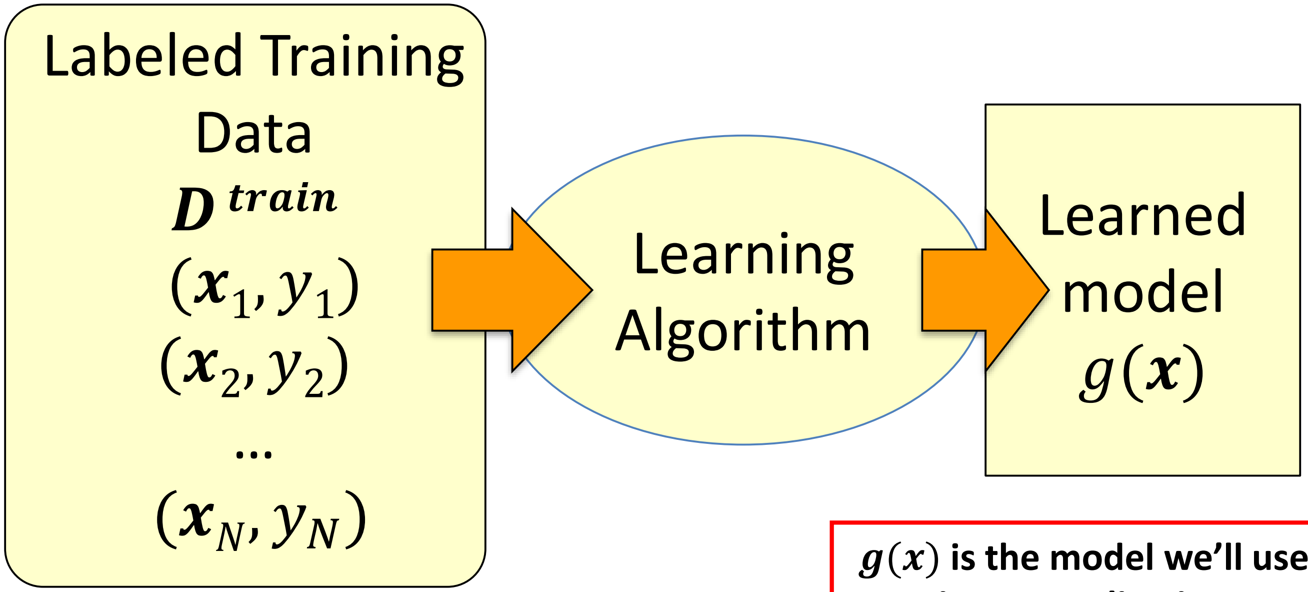
We typically use machine learning when the function $f(x)$ we want the system to apply is unknown to us, and we cannot "think" about it. The function could actually be simple.

Supervised Learning



Supervised learning: Training

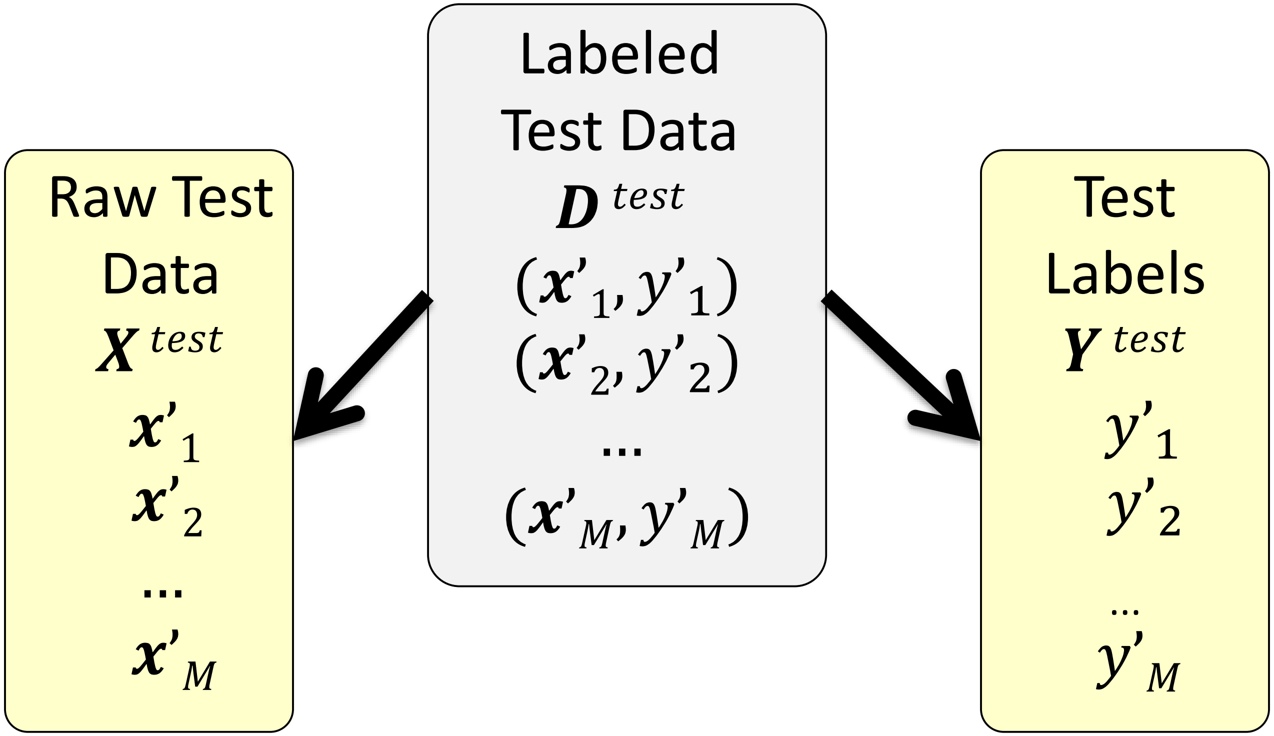
- Give the learner examples in $\boldsymbol{D}^{\,train}$.
- The learner returns a model $\mathrm{g}(x)$.



$g(x)$ is the model we'll use in our application

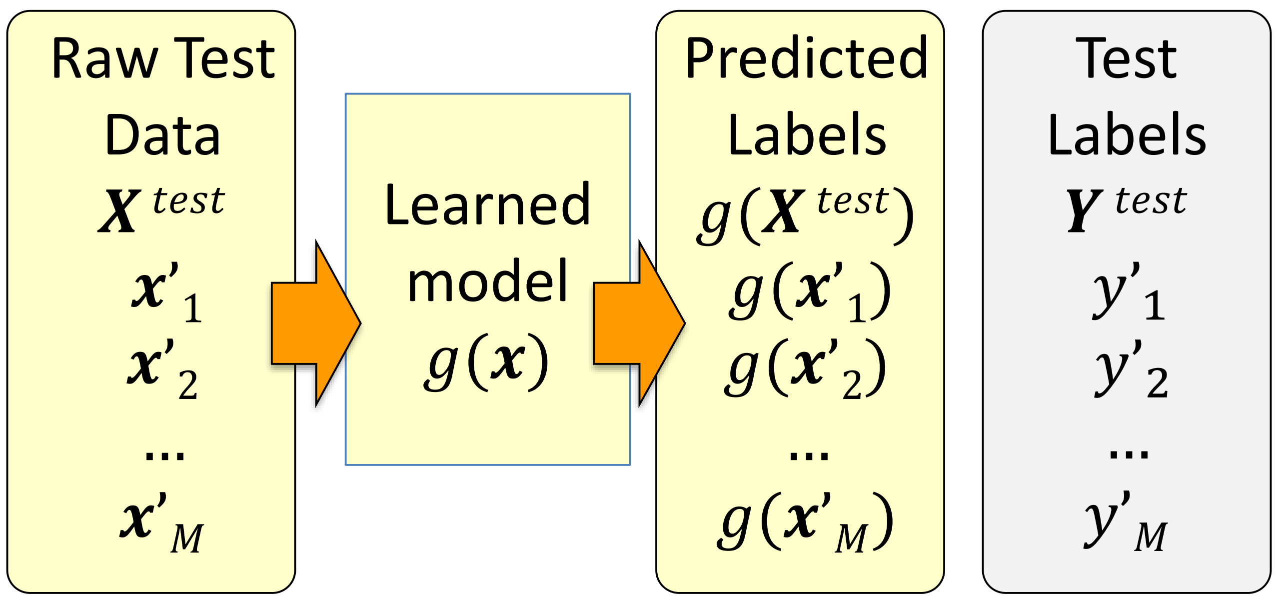
Supervised learning: Testing

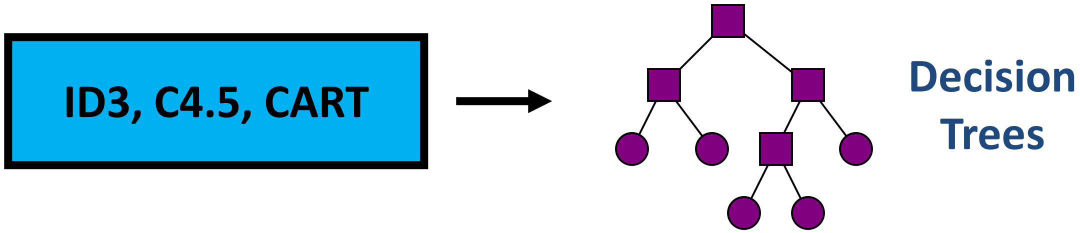
- Reserve some labeled data for testing



Supervised learning: Testing

- Apply the model to the raw test data.
- Evaluate by comparing predicted labels against the test labels.





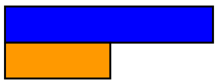
Decision Trees

An Introduction

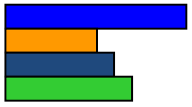
Representing Data

Think about a large table, N attributes, and assume you want to know something about the people represented as entries in this table.

E.g. own an expensive car or not.



- **Simplest way: Histogram on the first attribute – own**
- **Then, histogram on first and second (own & gender)**



But, what if the # of attributes is larger: N=16

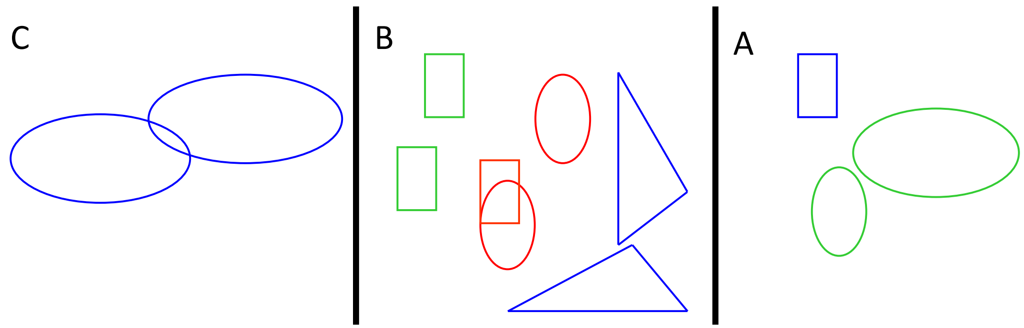
- **How large are the 1-d histograms *(contingency tables)*? .. 16 numbers**
- **How large are the 2-d histograms?.. 16-choose-2 = 120 numbers**
- **How large are the 3-d tables? .. 560 numbers**
- **With *<u>100 attributes</u>*, the *<u>3-d tables</u>* need 161,700 numbers**

- We need to figure out a way to represent data in a better way, and figure out what are the important attributes to look at first.

- Information theory has something to say about it – we will use it to better represent the data.

Decision Trees

- A hierarchical data structure that represents data by implementing a divide and conquer strategy.
- Given a collection of examples, learn a decision tree that represents it.
- Use this representation to classify new examples.

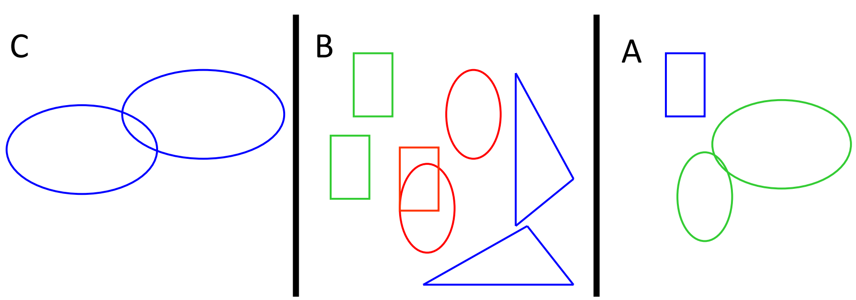


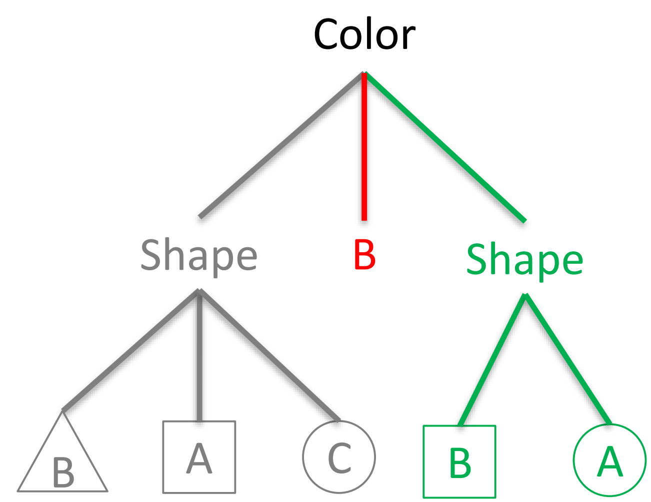
The Representation

Decision Trees are classifiers for instances represented as feature vectors

color={red, blue, green}; shape={circle, triangle, rectangle}; label= {A, B, C}

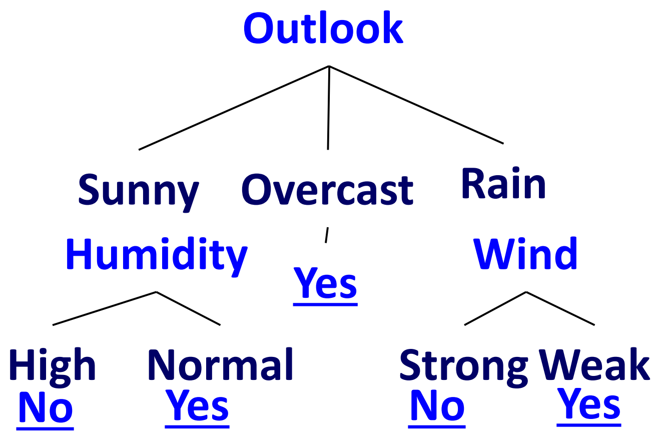
- **Nodes are tests for feature values**
- **There is one branch for each value of the feature**
- **Leaves specify the category (labels)**
- **Can categorize instances into multiple disjoint categories**





Decision Trees

- Can represent any Boolean Function.
- Can be viewed as a way to compactly represent a lot of data.
- The evaluation of the Decision Tree Classifier is easy.
- Clearly, given data, there are many ways to represent it as a decision tree.
- Learning a good representation from data is the challenge.



Will I play tennis today?

Features

Outlook:	{ Sun, Overcast, Rain }
Temperature:	{ Hot, Mild, Cool }
Humidity:	{ High, Normal, Low }
Wind:	{ Strong, Weak }

Labels

Binary classification task: Y = {+, -}

Will I play tennis today?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

Outlook: **S(unny), O(vercast), R(ainy)**

Temperature: **H(ot), M(edium), C(ool)**

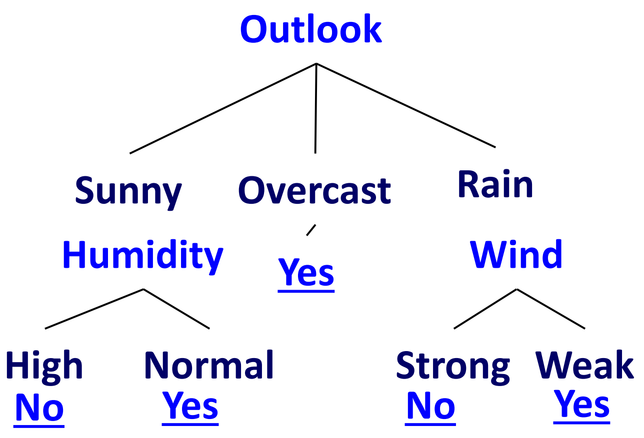
Humidity: **H(igh), N(ormal), L(ow)**

Wind: **S(trong), W(eak)**

Basic Decision Trees Learning Algorithm

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

- Data is processed in Batch (i.e. all the data available)
- Recursively build a decision tree top down.



Basic Decision Tree Algorithm

Let **S** be the set of Examples

  Label is the target attribute (the prediction)

  Attributes is the set of measured attributes

ID3(*S*, Attributes, Label)

If all examples are labeled the same return a single node tree with Label

  Otherwise Begin

  A = attribute in **Attributes** that ***best*** classifies **S** (Create a Root node for tree)

    for each possible value **v** of **A**

      Add a new tree branch corresponding to **A=v**

    Let ***Sv*** be the subset of examples in **S** with **A=v**

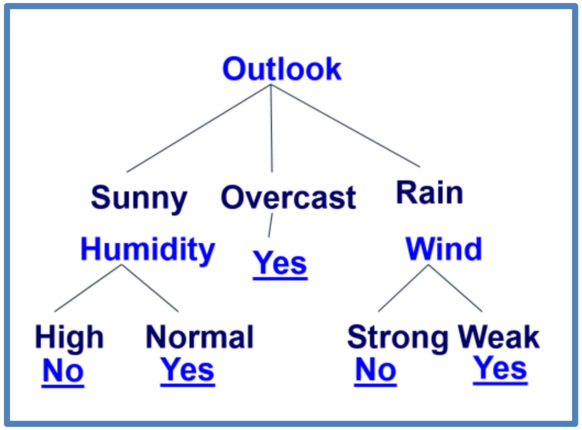
    if ***Sv*** is empty: add leaf node with the common value of **Label** in **S**

    Else: below this branch add the subtree

      ID3(*Sv*, Attributes - {a}, Label)

  End

  Return Root



Picking the Root Attribute

- The goal is to have the resulting decision tree as small as possible.
- The recursive algorithm is a greedy heuristic search for a simple tree, but cannot guarantee optimality.
- The main decision in the algorithm is the selection of the next attribute to condition on.

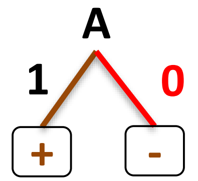
Picking the Root Attribute

Consider data with two Boolean attributes (A,B).

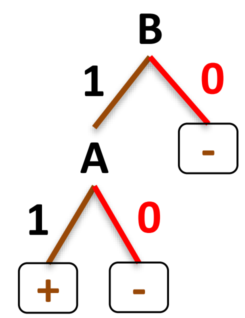
< (A=0,B=0), - >: 50 examples
< (A=0,B=1), - >: 50 examples
< (A=1,B=0), - >: 0 examples
< (A=1,B=1), + >: 100 examples

What should be the first attribute we select?

Splitting on A: we get purely labeled nodes.
Splitting on B: we don't get purely labeled nodes.
What if we have: <(A=1,B=0), - >: 3 examples?



splitting on A



splitting on B

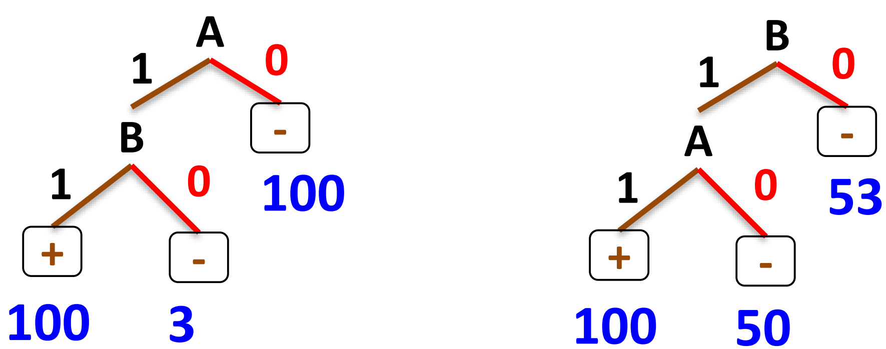
(one way to think about it: # of queries required to label a random data point)

Picking the Root Attribute

Consider data with two Boolean attributes (A,B).

< (A=0,B=0), - >: 50 examples
< (A=0,B=1), - >: 50 examples
< (A=1,B=0), - >: 3 examples
< (A=1,B=1), + >: 100 examples

What should be the first attribute we select?
Trees looks structurally similar; which attribute should we choose?



splitting on A   splitting on B

One way to think about it: # of queries required to label a random data point. If we choose A we have less uncertainty about the labels.

Picking the Root Attribute

- We want attributes that split the examples to sets that are relatively pure in one label; this way we are closer to a leaf node.
- The most popular heuristics is based on information gain, originated with the ID3 system of Quinlan.

Entropy

Entropy *(impurity, disorder)* of a set of examples, S, relative to a binary classification is:

$$Entropy(S) = -p_+ \log(p_+) - p_- \ log(p_-)$$

p_+ **is the proportion of positive examples in S and**

p_- **is the proportion of negatives examples in S**

If all the examples belong to the same category: Entropy = 0

If all the examples are equally mixed (0.5, 0.5): Entropy = 1

Entropy = Level of uncertainty.

In general, when p_i is the fraction of examples labeled i:

$$Entropy(S[p_1, p_2, \dots, p_k]) = -\sum_1^k p_i \, log(p_i)$$

Entropy can be viewed as the number of bits required, on average, to encode the class of labels. If the probability for + is 0.5, a single bit is required for each example; if it is 0.8 – can use less then 1 bit.

Entropy

Entropy (impurity, disorder) of a set of examples, S, relative to a binary classification is:

$$\boldsymbol{Entropy(S) = -p_+ \log(p_+) - p_- \log(p_-)}$$

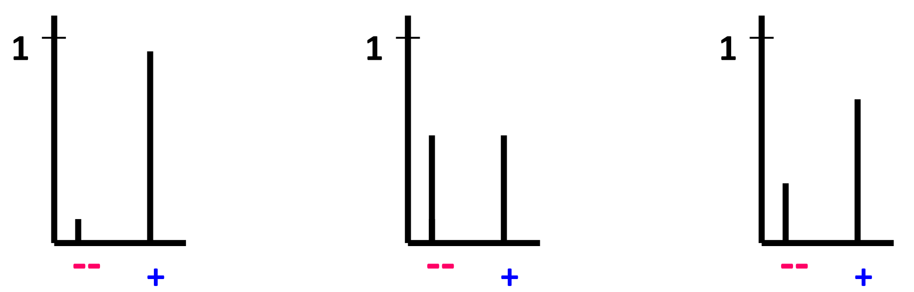
$\boldsymbol{p_+}$ **is the proportion of positive examples in S and**

$\boldsymbol{p_-}$ **is the proportion of negatives examples in S**

If all the examples belong to the same category: Entropy = 0

If all the examples are equally mixed (0.5, 0.5): Entropy = 1

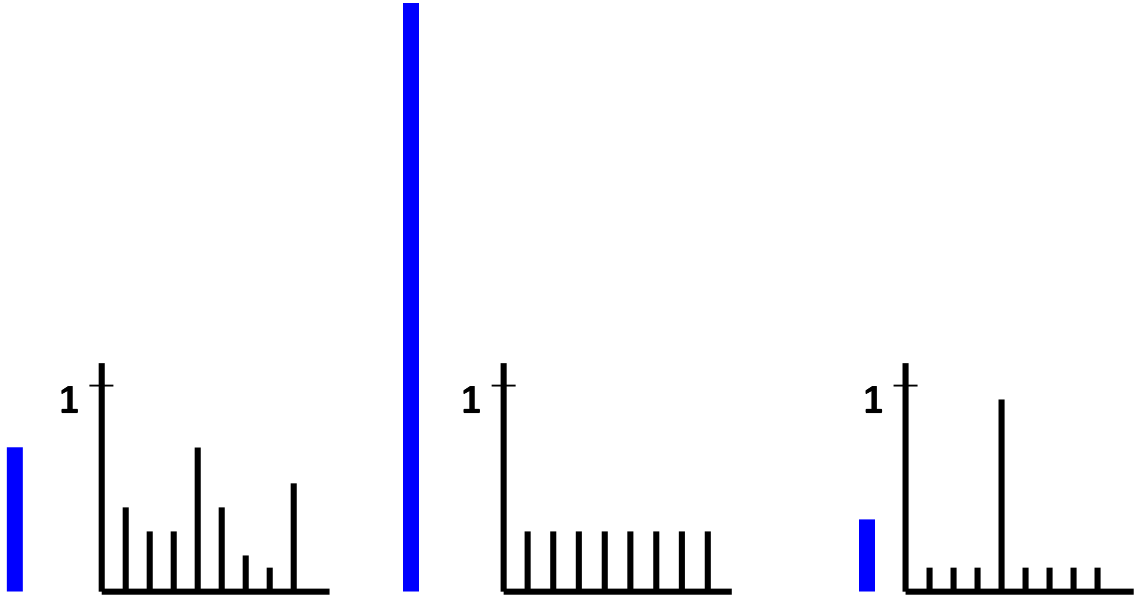
Entropy = Level of uncertainty.



Entropy

- The max value would be $\boldsymbol{log(k)}$
- Also note that the base of the log only introduce a constant factor; therefore, we'll think about base 2

$$Entropy(S[p_1, p_2 \ , \ldots, p_k]) = -\sum_{1}^{k} p_i \log(p_i)$$



Information Gain

High Entropy – High level of Uncertainty

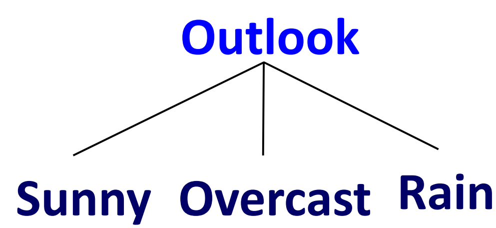
Low Entropy – No Uncertainty.

The information gain of an attribute a is the expected reduction in entropy caused by partitioning on this attribute

$$Gain(S,a) = Entropy(S) - \sum_{v \in values(S)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Where:

S_v is the subset of S for which attribute a has value v, and the entropy of partitioning the data is calculated by weighing the entropy of each partition by its size relative to the original set.



Partitions of low entropy *(imbalanced splits)* lead to high gain

Go back to check which of the A, B splits is better.

Will I play tennis today?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

Outlook: S(unny), O(vercast), R(ainy)

Temperature: H(ot), M(edium), C(ool)

Humidity: H(igh), N(ormal), L(ow)

Wind: S(trong), W(eak)

Will I play tennis today?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

Calculate current entropy

- $p_+ = \frac{9}{14}$ $\qquad p_- = \frac{5}{14}$
- $Entropy(Play) =$

$$-p_+ \log_2(p_+) - p_- \log_2(p_-)$$

$$= -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14}$$

$\approx$ **0.94**

Information Gain: Outlook

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

$$Gain(S,a) = Entropy(S) - \sum_{v \in values(S)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Outlook = sunny:

$p_+ = 2/5 \quad p_- = 3/5$ **Entropy(O = S) = 0.971**

Outlook = overcast:

$p_+ = 4/4 \quad p_- = 0$ **Entropy(O = O) = 0**

Outlook = rainy:

$p_+ = 3/5 \quad p_- = 2/5$ **Entropy(O = R) = 0.971**

Expected entropy

= $\sum_{v \in values(S)} \frac{|S_v|}{|S|} Entropy(S_v)$

= (5/14)×0.971 + (4/14)×0 + (5/14)×0.971 = 0.694

Information gain = 0.940 – 0.694 = 0.246

Information Gain: Humidity

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

$$Gain(S,a) = Entropy(S) - \sum_{v \in values(S)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Humidity = high:

$p_+ = 3/7 \quad p_- = 4/7$ **Entropy(H = H) = 0.985**

Humidity = Normal:

$p_+ = 6/7 \quad p_- = 1/7$ **Entropy(H = N) = 0.592**

Expected entropy

= $\sum_{v \in values(S)} \frac{|S_v|}{|S|} Entropy(S_v)$

= (7/14)×0.985 + (7/14)×0.592 = 0.7785

Information gain = 0.940 – 0.694 = 0.246

Which feature to split on?

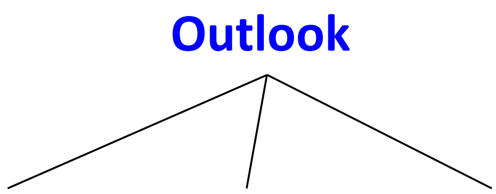
	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

Information gain:

Outlook:	0.246
Humidity:	0.151
Wind:	0.048
Temperature:	0.029

→ Split on Outlook

An Illustrative Example



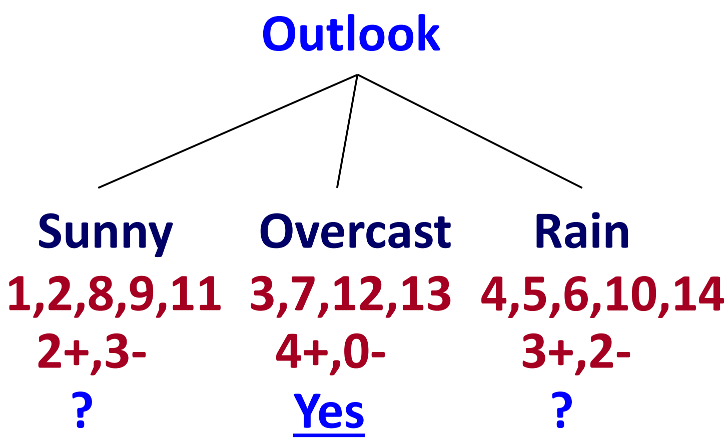
Gain(S,Humidity)=0.151

Gain(S,Wind) = 0.048

Gain(S,Temperature) = 0.029

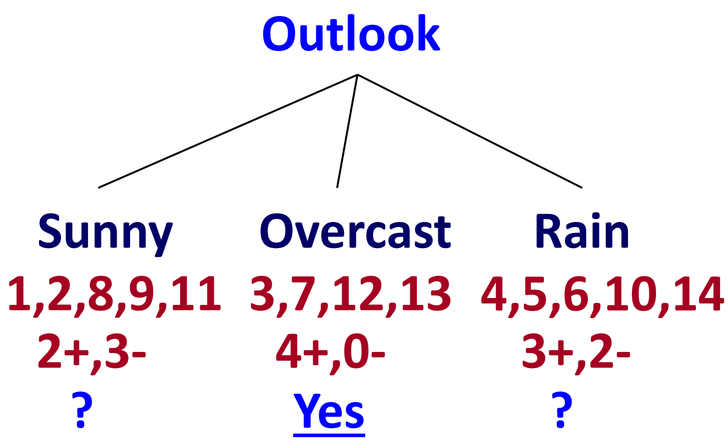
Gain(S,Outlook) = 0.246

An Illustrative Example



	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

An Illustrative Example

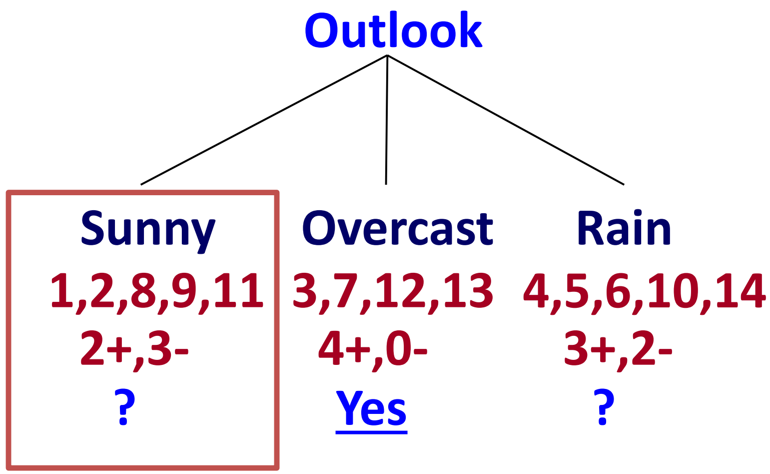


	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

Continue until:

- Every attribute is included in path, or,
- All examples in the leaf have same label.

An Illustrative Example



	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

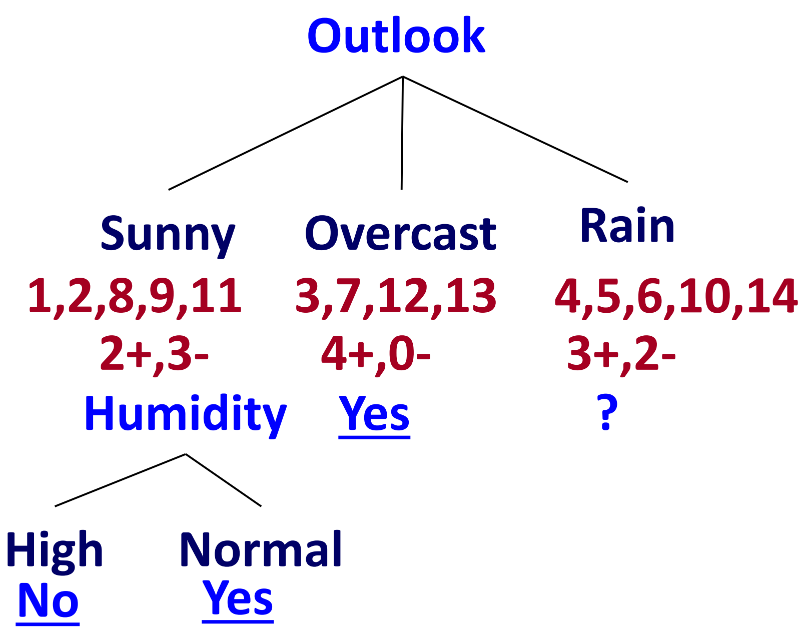
$\text{Gain}(S_{sunny}, \text{Humidity}) =$ .97-(3/5) 0-(2/5) 0 = .97

$\text{Gain}(S_{sunny}, \text{Temp}) =$ .97- 0-(2/5) 1 = .57

$\text{Gain}(S_{sunny}, \text{Wind}) =$ .97-(2/5) 1 - (3/5) .92= .02

Split on Humidity

An Illustrative Example



Function: induce_Decision_Tree(*S*)

1. Does S uniquely define a class?

if all $s \in S$ have the same label y: return S;

2. Find the feature with the most information gain:

$i = \text{argmax}_i \; \textit{Gain}(S, X_i)$

3. Add children to S:

for *k* in Values(X_i):

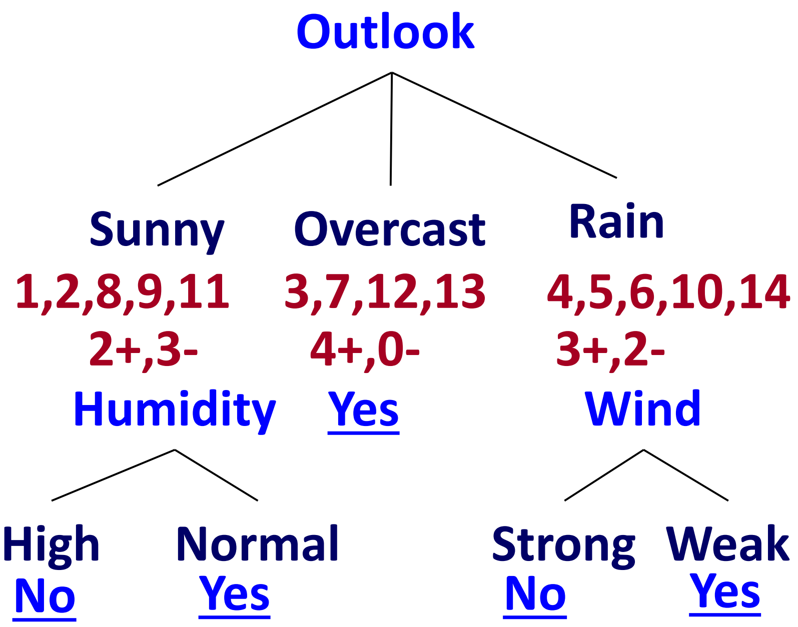
$S_k = \{s \in S \mid x_i = k\}$

addChild(S, S_k)

induceDecisionTree(S_k)

return S;

An Illustrative Example



Decision Trees Induction

- Conduct a search of the space of decision trees which can represent all possible discrete functions.
- Goal: to find the best decision tree
  - Best could be “smallest depth”
  - Best could be “minimizing the expected number of tests”
- Performs a greedy heuristic search: hill climbing without backtracking.
- Makes statistically based decisions using all data.

History of Decision Trees Research

- Hunt and colleagues in Psychology used full search decision tree methods to model human concept learning in the 60s
  - Quinlan developed ID3, with the information gain heuristics in the late 70s to learn expert systems from examples.
  - Breiman, Freidman and colleagues in statistics developed CART (classification and regression trees simultaneously).
- A variety of improvements in the 80s: coping with noise, continuous attributes, missing data, non-axis parallel etc.
  - Quinlan's updated algorithm, C4.5 (1993) is commonly used (New: C5).
- Boosting (or Bagging) over DTs is a very good general purpose algorithm.

Business Intelligence| *Business Analytics*

What is Business Intelligence (BI)?

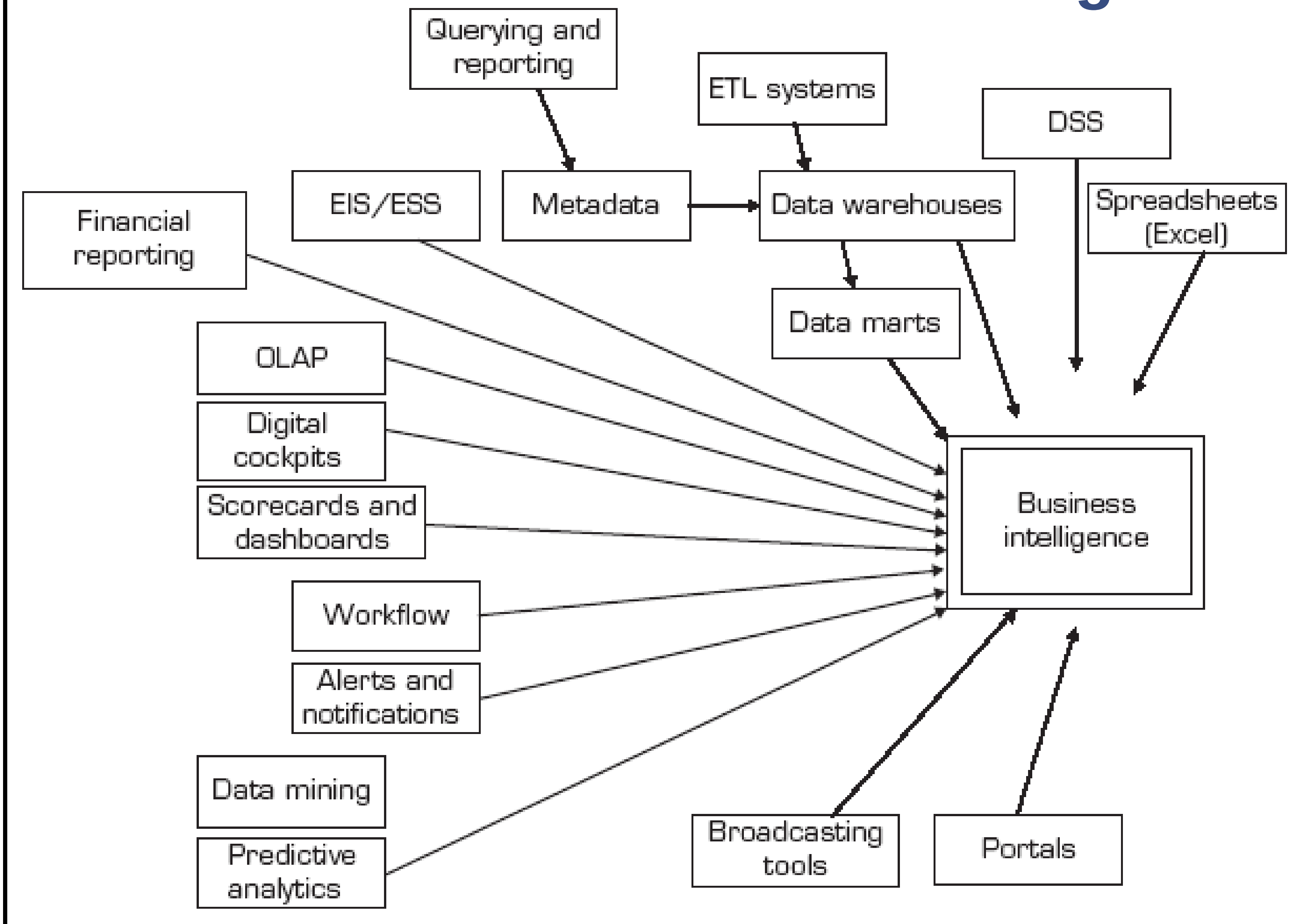
“ .. A segment of information technology that comprises software systems that enable ***finding, storing, organising*** and ***supplying data***; when incorporated into an information system, it enables company to utilise real-time analysis of information.”

“ .. Software that enables business users to ***see*** and ***use large amounts of complex data*** *(e.g. multidimensional analysis, query tools, data mining tools).”*

Why BI? .. Business Management Issues

- “We have mountains of data in this company, but we can’t access it.”
- “We need to slice and dice the data every which way.”
- “You’ve got to make it easy for business people to get at the data directly.”
- “Just show me what is important.”
- “It drives me crazy to have two people present the same business metrics at a meeting, but with different numbers.”
- “We want people to use information to support more fact-based decision making.”

A Framework for Business Intelligence



BI's Architecture & Components ..

Includes a Business Analytics Component ..

Business Analytics;

A collection of tools for manipulating, mining, and analyzing the data in the data warehouse;

- Create on-demand reports and queries and analyze data (originally called Online Analytical Processing – OLAP)
- **Automated decision systems**
- **Data Mining:** looks for hidden patterns in a collection of data which can be used to predict future behavior.

WEKA
The University of Waikato

Machine Learning Software in Java

Weka;
Waikato Environment for Knowledge Analysis

Machine Learning Software in Java ..

Waikato Environment for Knowledge Analysis (Weka)

WEKA
The University
of Waikato

- **Weka is a suite of machine learning software written in Java, developed at the University of Waikato, in Hamilton - New Zealand. It is free software licensed under the GNU General Public License.**

- **Weka is a collection of machine learning algorithms for data mining tasks. It contains tools for data preparation, classification, regression, clustering, association rules mining, and visualization.**

Machine Learning Software in Java ..

Waikato Environment for Knowledge Analysis (Weka)

WEKA
The University
of Waikato

- **Downloading and installing Weka ..**

https://www.cs.waikato.ac.nz/ml/weka/downloading.html

- **Free online courses on data mining with machine learning techniques in Weka ..**

https://www.cs.waikato.ac.nz/ml/weka/courses.html

Business Intelligence/Analytics Case-Study:

The German Credit Data Analysis

(Statlog Dataset)

Business Intelligence/Analytics (Finance/Banking Sector)

The Credit Risk Assessment Problem

Description: The business of banks is making loans. Assessing the credit worthiness of an applicant is of crucial importance. When a bank receives a loan application, based on the applicant's profile the bank has to make a decision regarding whether to go ahead with the loan approval or not. A bank's business rules regarding loans must consider two opposing factors. On the one hand, a bank wants to make as many loans as possible. Interest on these loans is the banks profit source. On the other hand, a bank can not afford to make too many bad loans. Too many bad loans could lead to the collapse of the bank. The bank's loan policy must involve a compromise. Not too strict and not too lenient.

In other words, two types of risks are associated with the bank's decision:

1. If the applicant is a good credit risk, i.e. is likely to repay the loan, then not approving the loan to the person results in a loss of business to the bank.
2. If the applicant is a bad credit risk, i.e. is not likely to repay the loan, then approving the loan to the person results in a financial loss to the bank.

So, it's on the part of the bank or other lending authority to evaluate the risks associated with lending money to a customer. You have to develop a system to help a loan officer decide whether the credit of a customer is good or bad.

The German Credit Data Analysis

Abstract: This dataset *(consisting of 1000 actual cases collected in Germany)* classifies people described by a set of attributes as good or bad credit risks. The dataset contains data on 20 variables and the classification of whether an applicant is considered a Good or a Bad credit risk for 1000 loan applicants.

The German Credit data set is a publicly available data set downloaded from the UCI Machine Learning Repository:
https://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29

- Data Set Characteristics (**Multivariate**) – Number of Instances (**1000**)
- Area (**Financial**) – Attribute Characteristics (**Categorical, & Integer**)
- Number of Attributes (**20**) – Date Donated (**1994-11-17**)
- Associated Tasks (**Classification**) – Missing Values (**N/A**)

Source:
Professor Dr. Hans Hofmann
Institut für Statistik & Ökonometrie - Universität Hamburg
FB Wirtschaftswissenschaften
Von-Melle-Park 5
2000 Hamburg 13

Attributes Information *(Overall)*

Attribute 1: (qualitative) Status of existing checking account
Attribute 2: (numerical) Duration in month
Attribute 3: (qualitative) Credit history
Attribute 4: (qualitative) Purpose
Attribute 5: (numerical) Credit amount
Attribute 6: (qualitative) Savings account/bonds
Attribute 7: (qualitative) Present employment since
Attribute 8: (numerical) Installment rate in percentage of disposable income
Attribute 9: (qualitative) Personal status and sex
Attribute 10: (qualitative) Other debtors / guarantors
Attribute 11: (numerical) Present residence since
Attribute 12: (qualitative) Property
Attribute 13: (numerical) Age in years
Attribute 14: (qualitative) Other installment plans
Attribute 15: (qualitative) Housing
Attribute 16: (numerical) Number of existing credits at this bank
Attribute 17: (qualitative) Job
Attribute 18: (numerical) Number of people being liable to provide maintenance for
Attribute 19: (qualitative) Telephone
Attribute 20: (qualitative) Foreign worker

Attributes Information

(Sample of the Values for Selected Attributes)

Attribute 1: (qualitative) Status of existing checking account

A11 : ... < 0 DM

A12 : 0 <= ... < 200 DM

A13 : ... >= 200 DM / salary assignments for at least 1 year

A14 : no checking account

Attribute 4: (qualitative) Purpose

A40 : car (new)

A41 : car (used)

A42 : furniture/equipment

A43 : radio/television

A44 : domestic appliances

A45 : repairs

A46 : education

A47 : (vacation - does not exist?)

A48 : retraining

A49 : business

A410 : others

Attribute 17: (qualitative) Job

A171 : unemployed / unskilled - non-resident

A172 : unskilled - resident

A173 : skilled employee / official

A174 : management / self-employed / highly qualified employee

Attribute 20: (qualitative) Foreign worker

A201 : yes

A202 : no

Thanks! . . . *Questions?*