Lecture 4: Decision Tree

Winter 2018

Kai-Wei Chang

CS @ UCLA

kw+cm146@kwchang.net

The instructor gratefully acknowledges Dan Roth, Sriram Sankararaman, Fei Sha, Ameet Talwalkar, Eric Eaton, and Jessica Wu whose slides are heavily used, and the many others who made their course material freely available online.

Key issues in machine learning

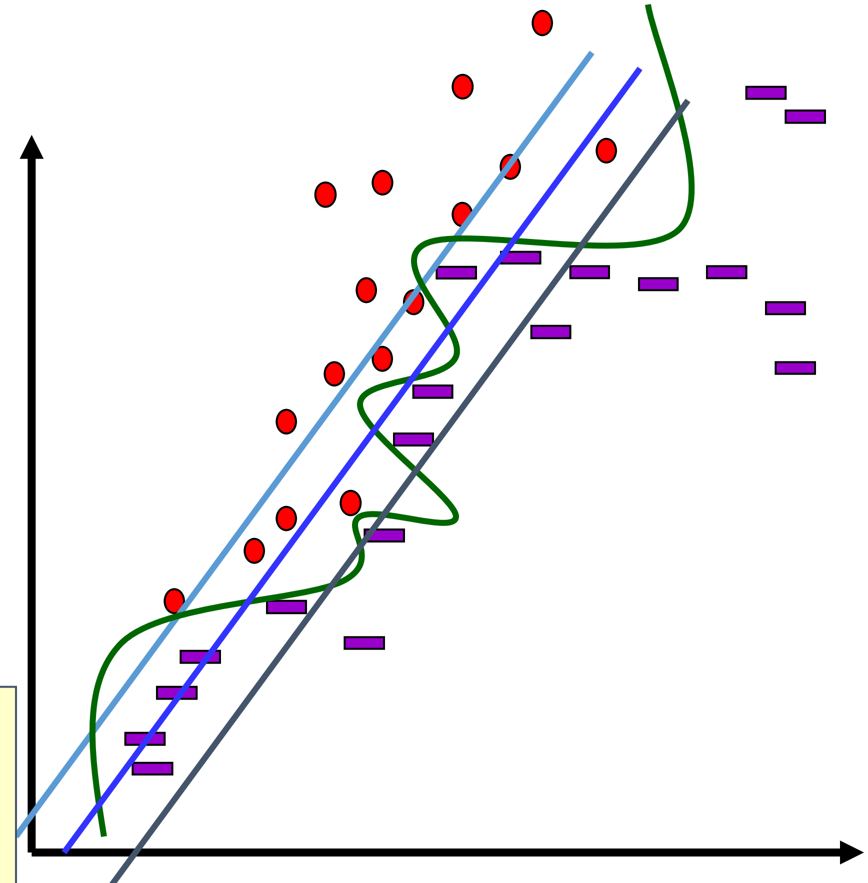
- Modeling
  - How to formulate your problem?
- Representation
  - What is the input/output space?
  - What is the hypothesis space?
- Algorithms
  - How to find the best hypothesis?

What Did We Learn?

- Learning problem:

  Find a function that best separates the data

- What function?
- What's best?
- How to find it?



Linear:

x= data representation;

w= the classifier

Y = sgn $\{w^T x\}$

- A possibility: Define the learning problem to be:

  Find a (linear) function that best separates the data

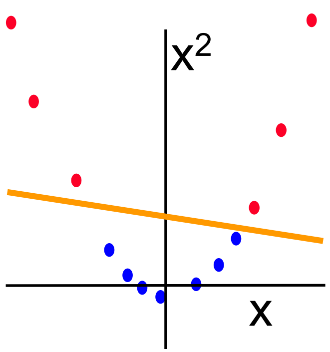
Motivation

- Question 1: Our solution learns a linear function; in principle, the target function may not be linear
  - **Can we learn a function that is more flexible in terms of what it does with the feature space?**
- Question 2: Can we say something about the quality of what we learn (sample complexity, time complexity; quality)

Decision Trees

- Earlier, we decoupled the generation of the feature space from the learning.
- Argued that we can map the examples into another space, in which the target functions are linearly separable.
- How do we determine what are good mappings?

- The study of decision trees may shed some light on this.
- Learning is done directly from the given data representation.
- The algorithm ``transforms" the data itself.



This Lecture

- Model/Representation: Decision trees
  - Non-linear classifiers

- Algorithm: Learning decision trees (ID3 algorithm)
  - Greedy heuristic (based on information gain)
    Originally developed for discrete features
  - Some extensions to the basic algorithm

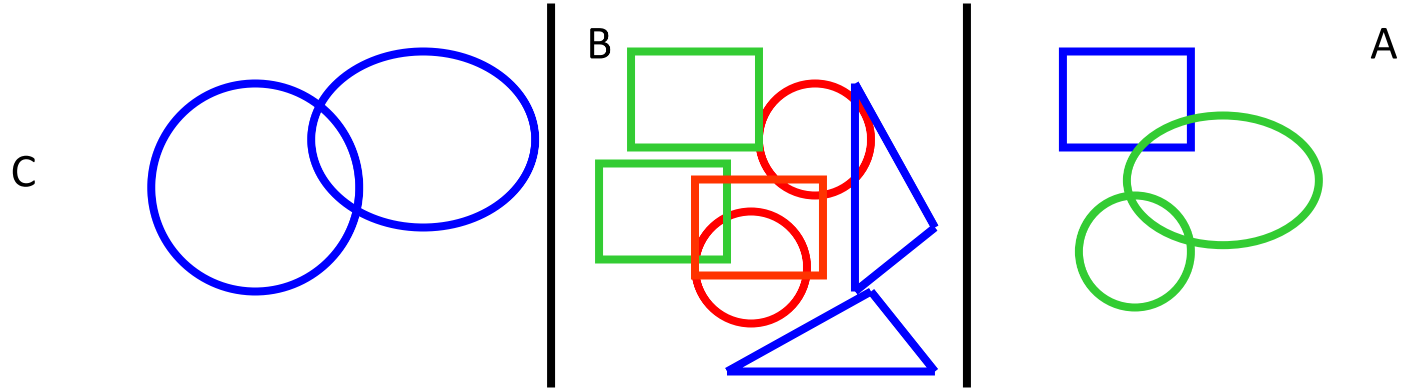
This Lecture

- Model/Representation: Decision trees
  - Non-linear classifiers

- Algorithm: Learning decision trees (ID3 algorithm)
  - Greedy heuristic (based on information gain)
    Originally developed for discrete features
  - Some extensions to the basic algorithm

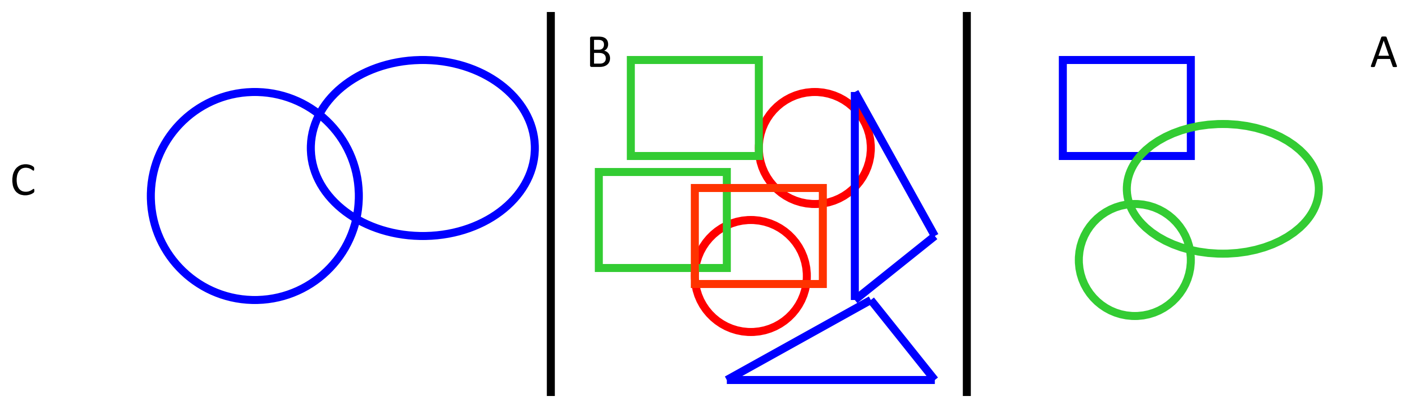
Sample dataset

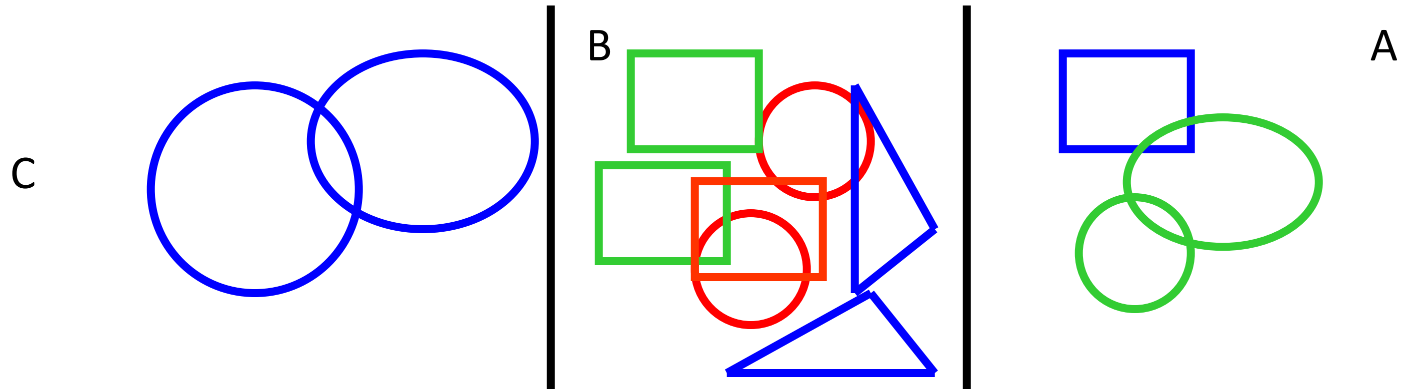
- What features we can used?
- What is the label for a red triangle?

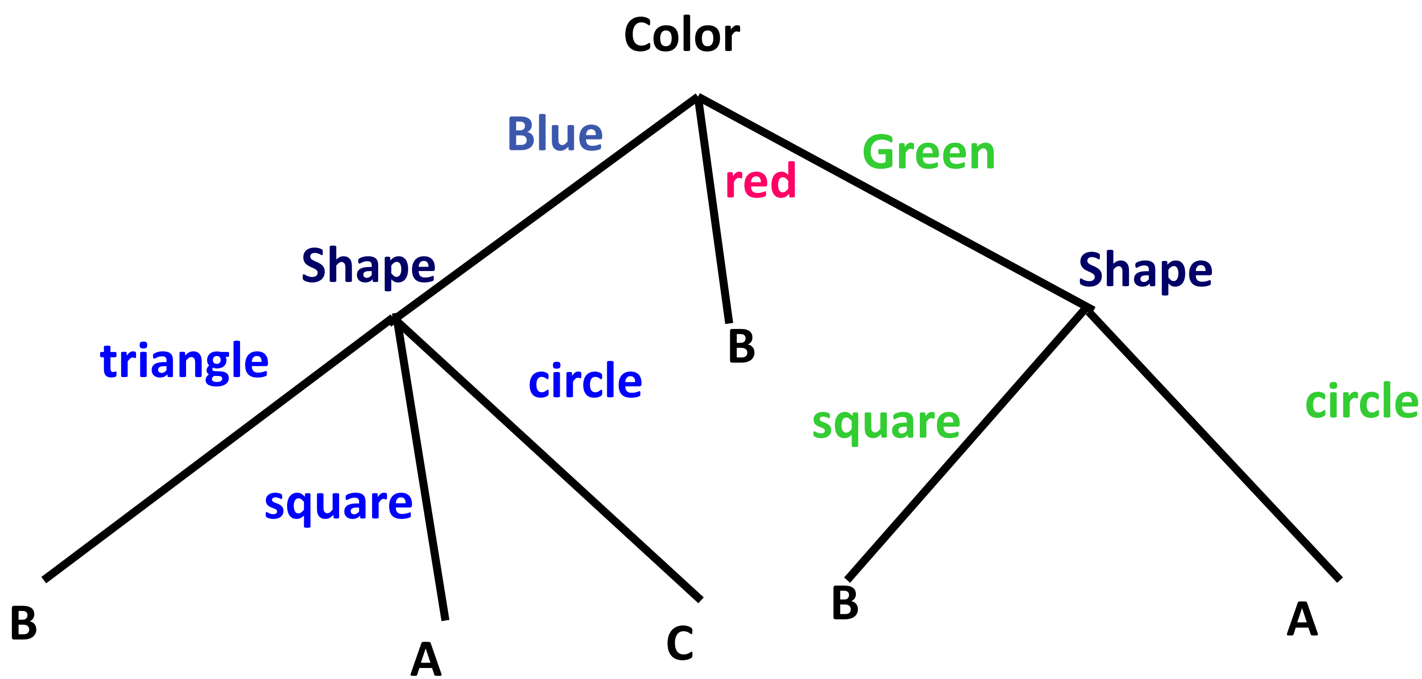


Decision Trees

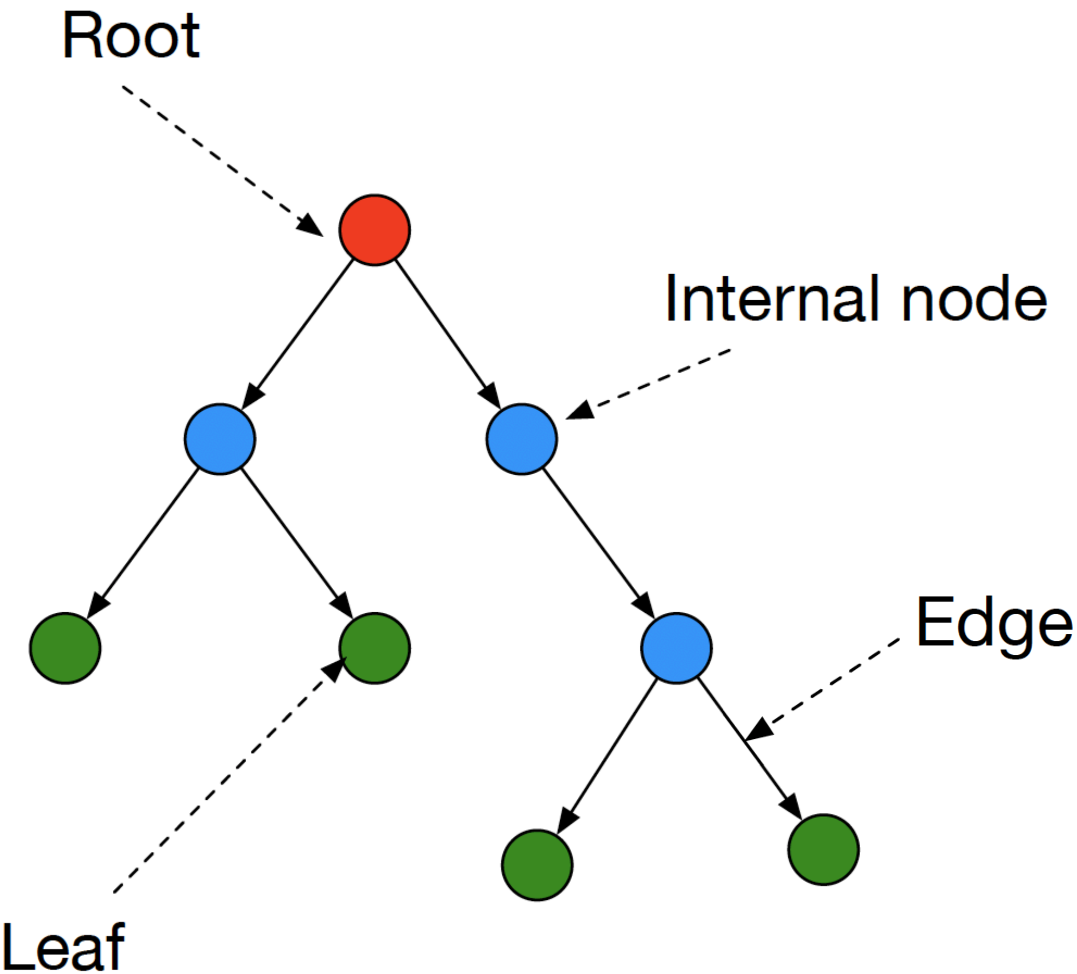
- A hierarchical data structure that represents data by implementing a divide and conquer strategy
- Can be used as a non-parametric classification and regression method
- Given a collection of examples, **learn a decision tree that represents it.**
- Use this representation to **classify new examples**




C
B
A


Color
Blue
red
Green
Shape
B
Shape
triangle
circle
square
circle
square
B
A
C
B
A

Terminology

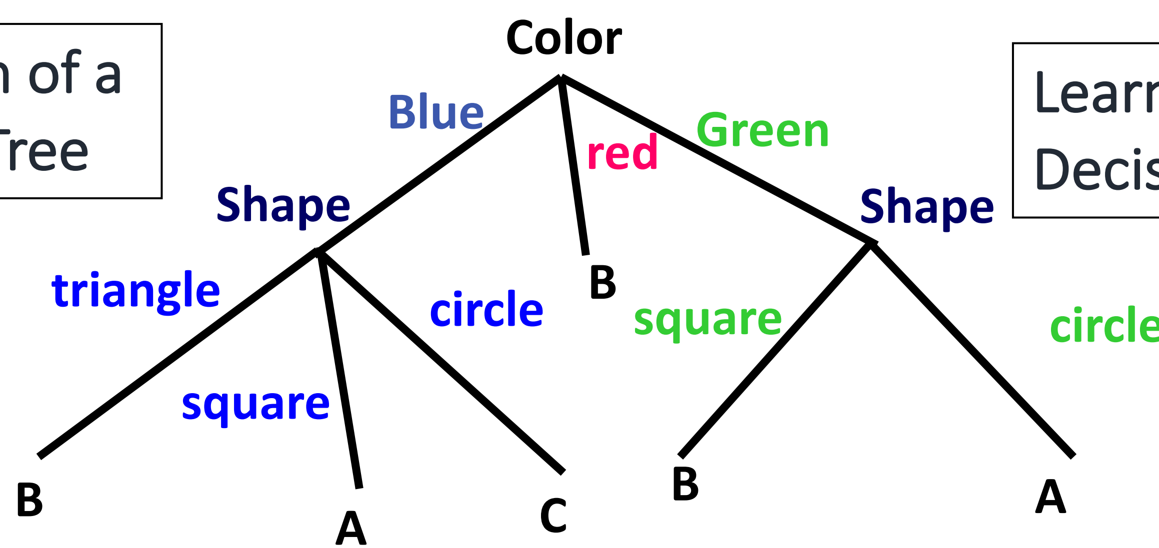


Will sometimes drop the arrows on the edges

The Representation

- Decision Trees are classifiers for instances represented as feature vectors (color= ; shape= ; label=)
- Nodes are tests for feature values
- Edges: There is one branch for each value of the feature
- Leaves specify the category (labels)
- Can categorize instances into multiple disjoint categories

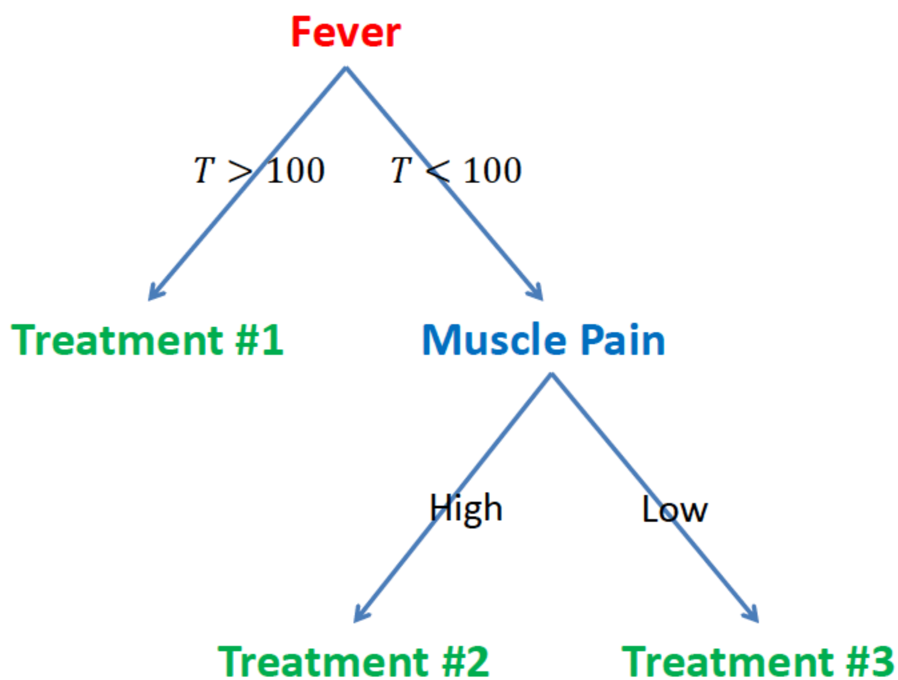
Evaluation of a Decision Tree



Learning a Decision Tree

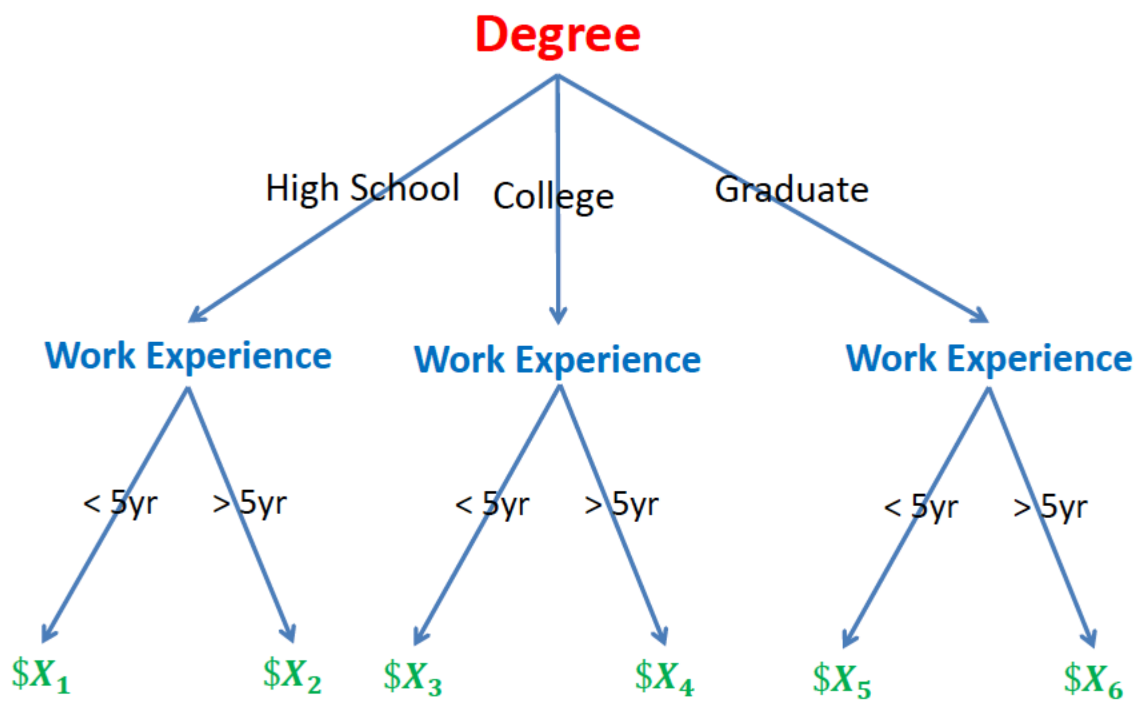
Motivations:
Many decisions are tree structures

Medical treatment



Motivations: Many decisions are tree structures

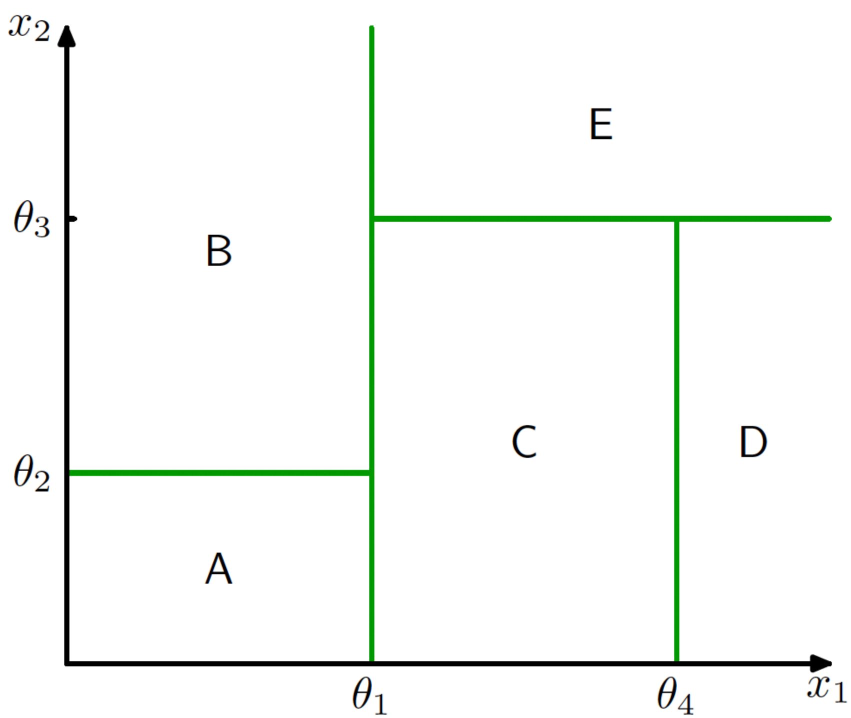
Salary in a company

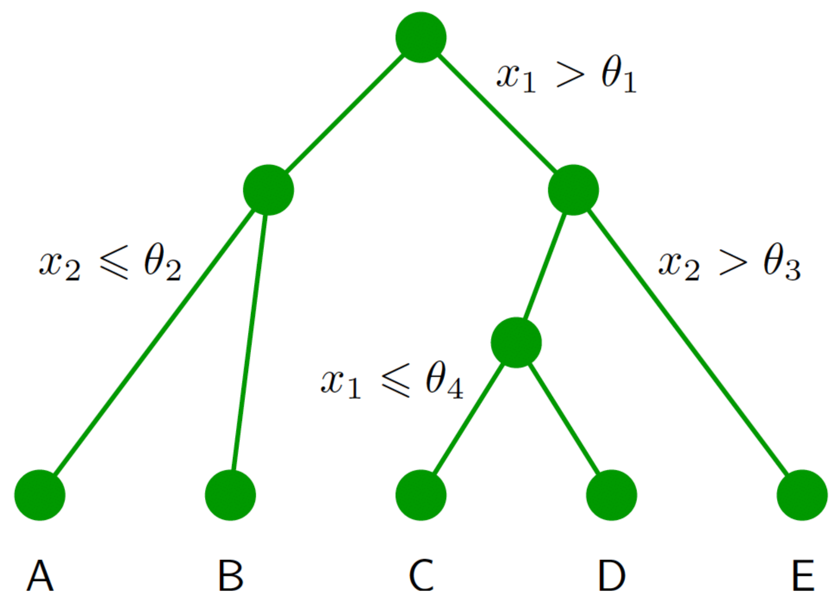


Decision Boundaries

- Usually, instances are represented as attribute-value pairs (color=blue, shape = square, +)
- Numerical values can be used either by discretizing or by using thresholds for splitting nodes
- In this case, the tree divides the features space into axis-parallel rectangles, each labeled with one of the labels

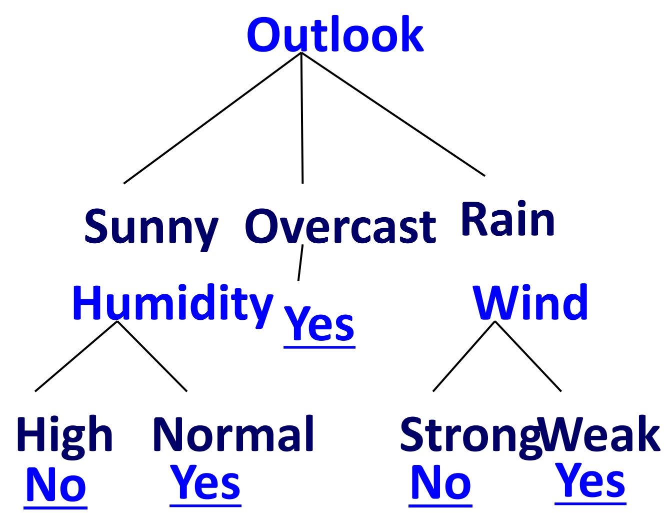
A tree partitions the feature space





Advantages of Decision tree

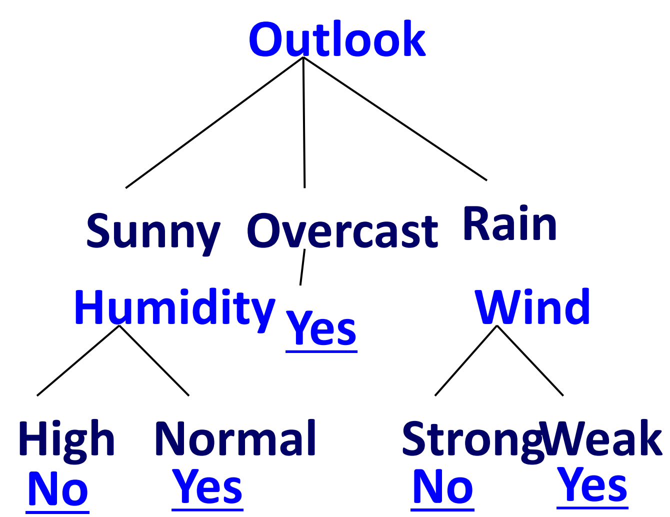
- Can represent any Boolean Function
- Can be viewed as a way to compactly represent a lot of data.
- Natural representation: (20 questions)
- The evaluation of the Decision Tree Classifier is easy



Challenge

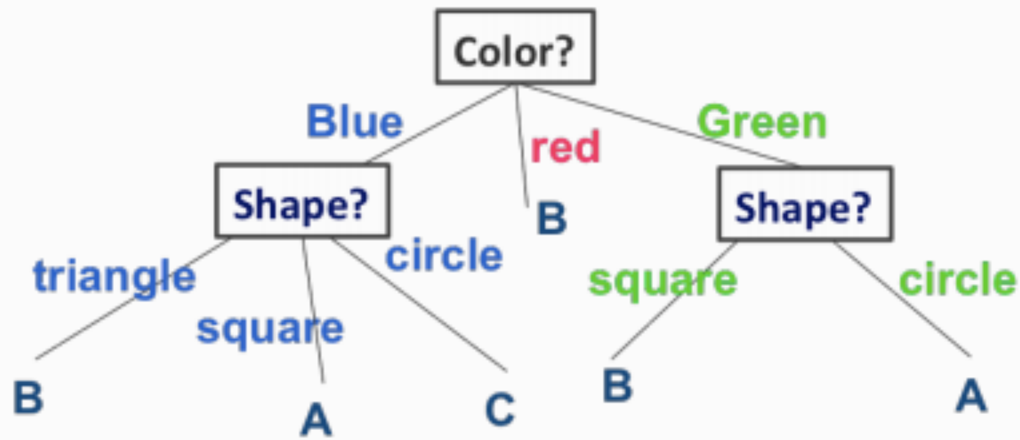
- Clearly, given data, there are many ways to represent it as a decision tree.
- Learning a good representation from data is the challenge.

A tree partitions the feature space



Expressivity of Decision Trees

❖ What Boolean functions can decision trees represent? -- any Boolean function



(Color=blue AND Shape=triangle ⇒ Label=B) AND

(Color=blue AND Shape=square ⇒ Label=A) AND

(Color=blue AND Shape=circle ⇒ Label=C) AND….

Decision Trees

- Output is a discrete category. Real valued outputs are possible (regression trees)
- There are **efficient** algorithms for processing large amounts of data (but not too many features)
- There are methods for handling **noisy data** (classification noise and attribute noise) and for handling **missing attribute values**

Learning a decision tree

This Lecture

- Model/Representation: Decision trees
  - Non-linear classifiers

- Algorithm: Learning decision trees (ID3 algorithm)
  - Greedy heuristic (based on information gain)
    Originally developed for discrete features
  - Some extensions to the basic algorithm

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)

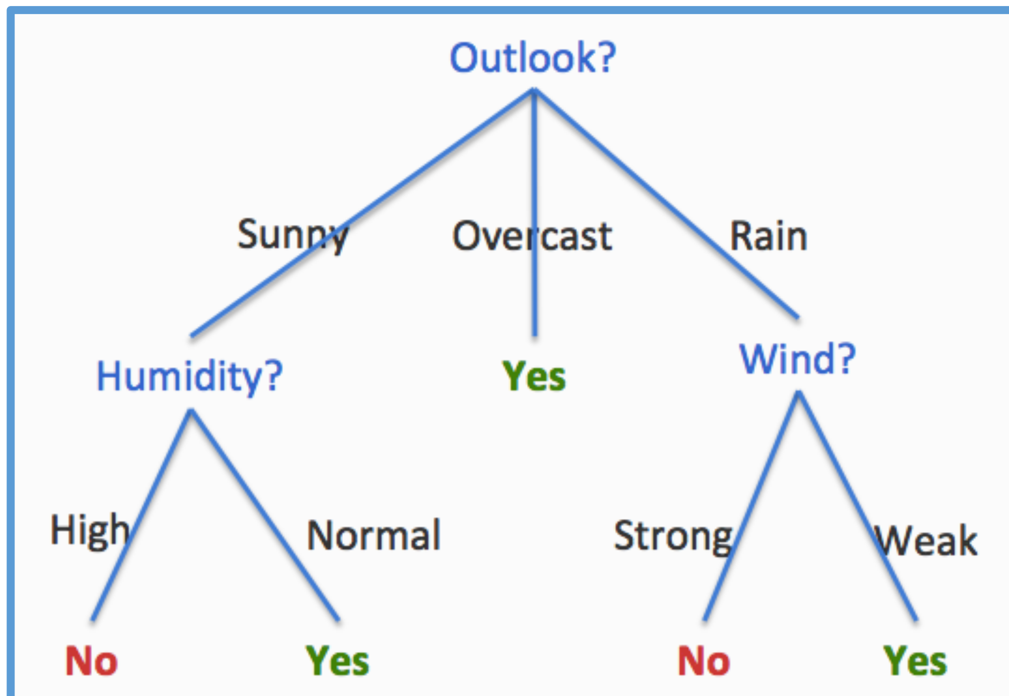
Humidity: H(igh), N(ormal), L(ow)

Wind: S(trong), W(eak)

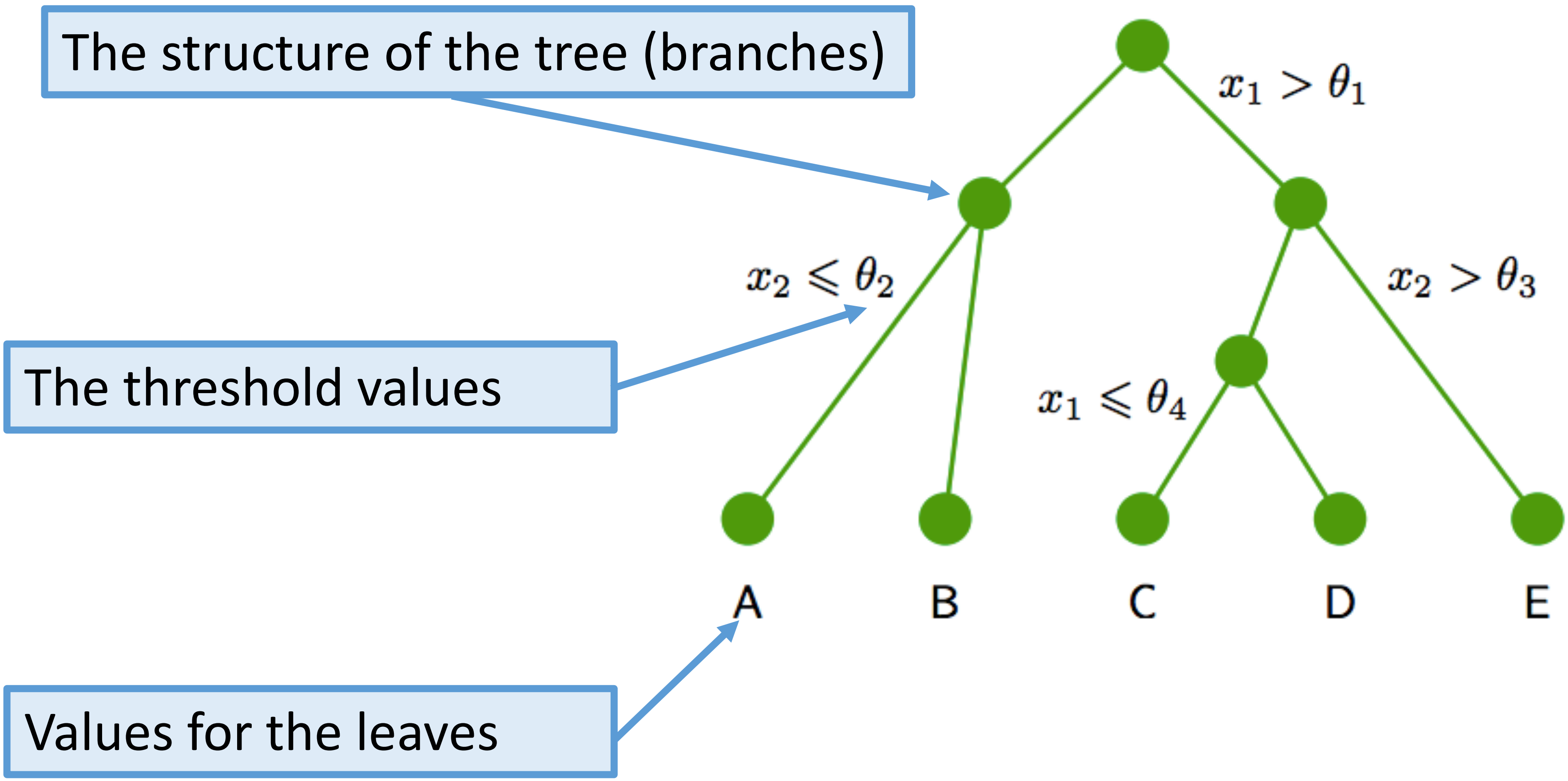
Basic Decision Trees Learning Algorithm

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

	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	-



What do we need to learn?

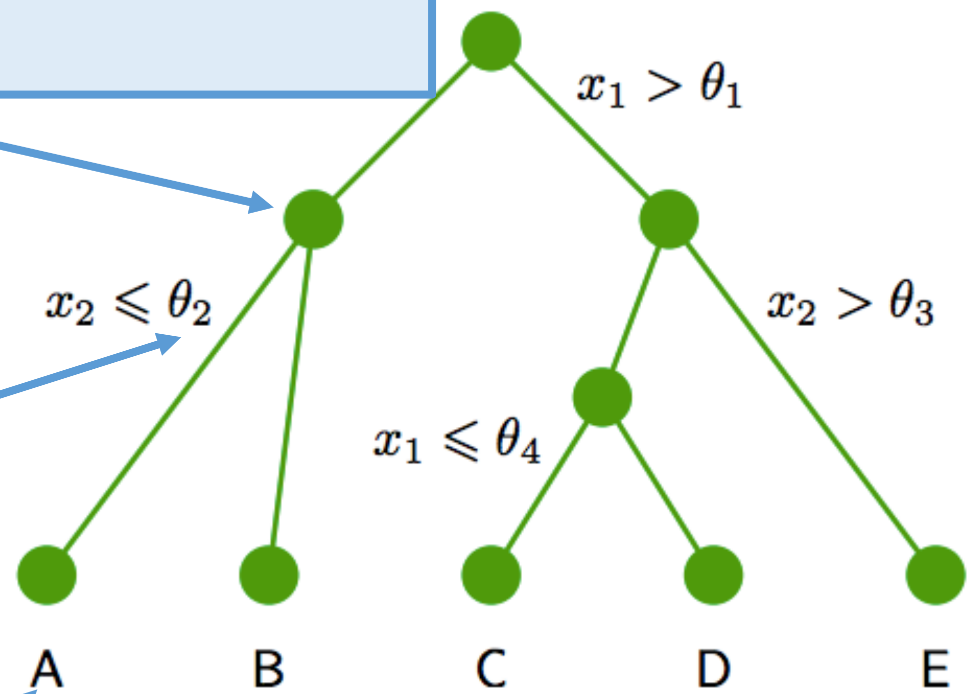


What do we need to learn?

Pick one feature that best classifier the data
How?

Create branches based on feature values

Values for the leaves = the output label.
When to stop?



DT algorithm: ID3(*S*, Attributes, Label)

1. If all examples have a same label
   return a single node tree with Label
2. Create a Root node for tree
3. A = attribute in Attributes that ***best*** classifies S
4. For each possible value v of A
   1. Add a new tree branch corresponding to A=v
   2. Let *Sv* be the subset of examples in S with A=v
   3. 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)
4. Return Root

DT algorithm: ID3(*S*, Attributes, Label)

- A recursive algorithm
- Recursively build a decision tree top down.
- Base case:

  If all examples are labeled the same

  Return a single node tree with Label

  Otherwise

  Recursive decision tree algorithm

  (see next slide)

Which attribute to split?

- The goal is to have the resulting decision tree as small as possible
  - But, finding the minimal decision tree consistent with the data is NP-hard
- 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.

Which attribute to split?

Patrons? is a better choice—gives **information** about the classification

Need a way to quantify things

Which attribute to split?

- The goal is to have the resulting decision tree as small as possible
- The main decision in the algorithm is the selection of the next attribute to condition on.
- 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.

How to measure information gain?

- Idea: Gaining information reduces uncertainty
- Uncertainty can be measured by Entropy

Vincent Van Gogh: Bedroom in Arles

High entropy

By Ursus Wehrli

Low entropy

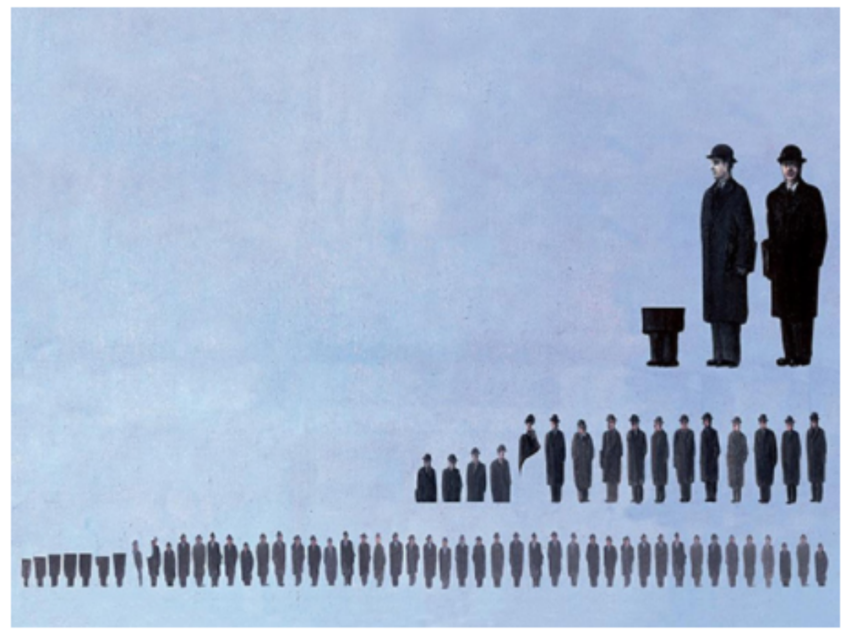
How to measure information gain?

- Idea: Gaining information reduces uncertainty
- Uncertainty can be measured by Entropy

René Magritte "Golconda"

High entropy

By Ursus Wehrli



Low entropy

Entropy

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

$$\mathbf{Entropy(S)} = -\mathbf{p}_{+}\mathbf{log(p}_{+}\mathbf{)} - \mathbf{p}_{-}\mathbf{log(p}_{-}\mathbf{)}$$

- where $\mathbf{P_+}$ is the proportion of positive examples in S and $\mathbf{P_-}$ is the proportion of negatives.
  - 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 (formal definition)

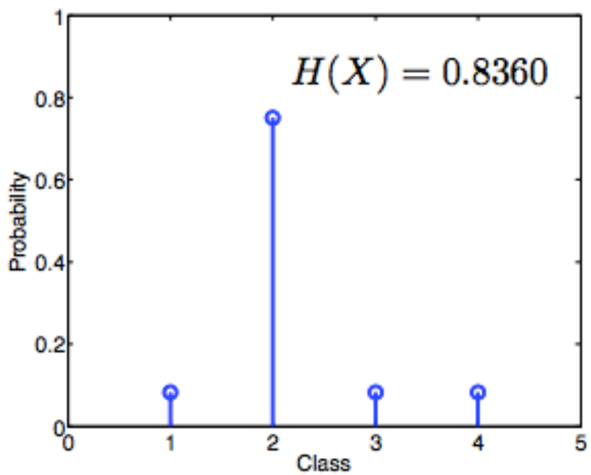
- If a random variable S has K different values, $a_1, a_2, \ldots a_K$, it is entropy is given by

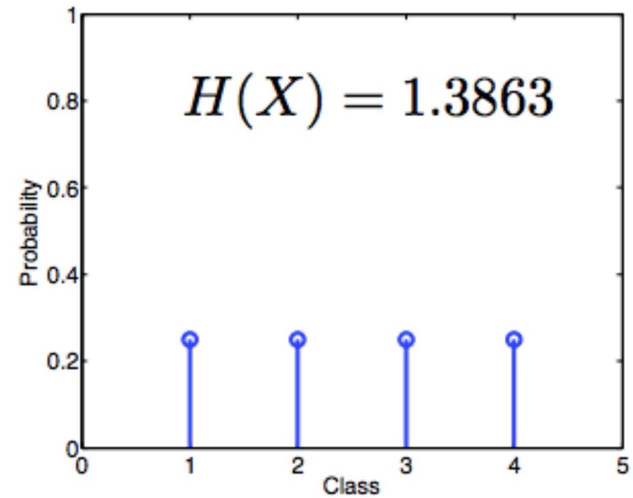
$$H[S] = -\sum_{v=1}^{K} P(S = a_v) \log P(S = a_v)$$

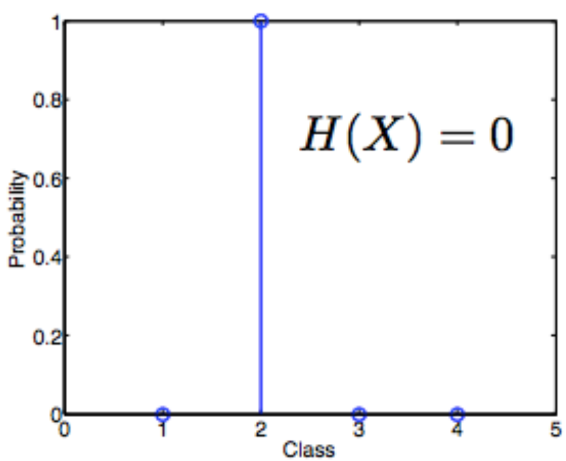
- Measures the amount of uncertainty of a random variable with a specific probability distribution. Higher it is, less confident we are in its outcome

Example

Entropy







Which attribute to split

Patrons? is a better choice—gives **information** about the classification

Patron vs. Type?

By choosing Patron, we end up with a partition (3 branches) with smaller entropy, ie, smaller uncertainty (0.45 bit)

By choosing Type, we end up with uncertainty of 1 bit.

Thus, we choose Patron over Type.

Uncertainty if we go with "Patron"

For "None" branch

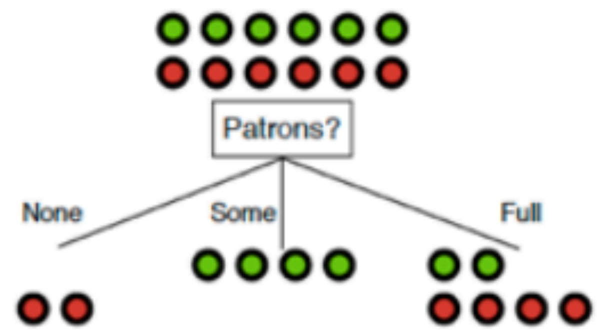
$$-\left(\frac{0}{0+2}\log\frac{0}{0+2}+\frac{2}{0+2}\log\frac{2}{0+2}\right)=0$$

For "Some" branch

$$-\left(\frac{4}{4+0}\log\frac{4}{4+0}+\frac{4}{4+0}\log\frac{4}{4+0}\right)=0$$

For "Full" branch

$$-\left(\frac{2}{2+4}\log\frac{2}{2+4}+\frac{4}{2+4}\log\frac{4}{2+4}\right)\approx 0.9$$



For choosing "Patrons"

weighted average of each branch: this quantity is called conditional entropy

$$\frac{2}{12}*0+\frac{4}{12}*0+\frac{6}{12}*0.9=0.45$$

Information Gain

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

$$Gain(S, A) = Entropy(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

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

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	-

Current entropy:

$p = 9/14$

$n = 5/14$

$H(\text{Play?}) = -(9/14)\log_2(9/14)$
$-(5/14)\log_2(5/14)$
≈ 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	-

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	-

Outlook = sunny: 5 of 14 examples

$p = 2/5$ $n = 3/5$ $\mathbf{H_S = 0.971}$

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	-

Outlook = sunny: 5 of 14 examples

$p = 2/5 \quad n = 3/5 \quad \mathbf{H_S = 0.971}$

Outlook = overcast: 4 of 14 examples

$p = 4/4 \quad n = 0 \quad \mathbf{H_o = 0}$

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	-

Outlook = sunny: 5 of 14 examples

$p = 2/5$ $n = 3/5$ $\mathbf{H_S = 0.971}$

Outlook = overcast: 4 of 14 examples

$p = 4/4$ $n = 0$ $\mathbf{H_O = 0}$

Outlook = rainy: 5 of 14 examples

$p = 3/5$ $n = 2/5$ $\mathbf{H_R = 0.971}$

Expected entropy:

$(5/14)\times 0.971 + (4/14)\times 0 + (5/14)\times 0.971$

$= \mathbf{0.694}$

Information gain:

$0.940 - 0.694 = \mathbf{0.246}$

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	-

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	-

Humidity = High:

$p = 3/7$ $n = 4/7$ $\mathbf{H_h = 0.985}$

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	-

Humidity = High:

$p = 3/7 \quad n = 4/7$ **$H_h = 0.985$**

Humidity = Normal:

$p = 6/7 \quad n = 1/7$ **$H_o = 0.592$**

Expected entropy:

$(7/14)\times0.985 + (7/14)\times0.592 =$ **0.7885**

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	-

Humidity = High:

$p = 3/7 \quad n = 4/7 \qquad \mathbf{H_h = 0.985}$

Humidity = Normal:

$p = 6/7 \quad n = 1/7 \qquad \mathbf{H_o = 0.592}$

Expected entropy:

$(7/14)\times0.985 + (7/14)\times0.592 = \mathbf{0.7885}$

Information gain:

$0.940 - 0.7885 = \mathbf{0.1515}$

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

Which feature to split on?

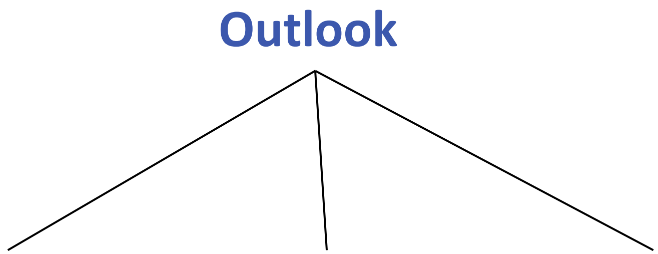
	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

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

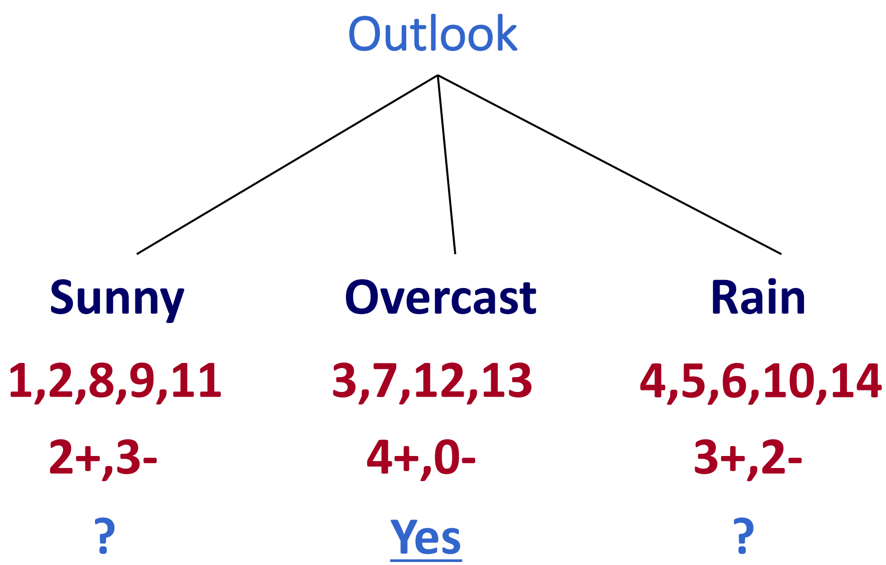
An Illustrative Example

Outlook

Sunny	Overcast	Rain
1,2,8,9,11	3,7,12,13	4,5,6,10,14
2+,3-	4+,0-	3+,2-
?	Yes	?

	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

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

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

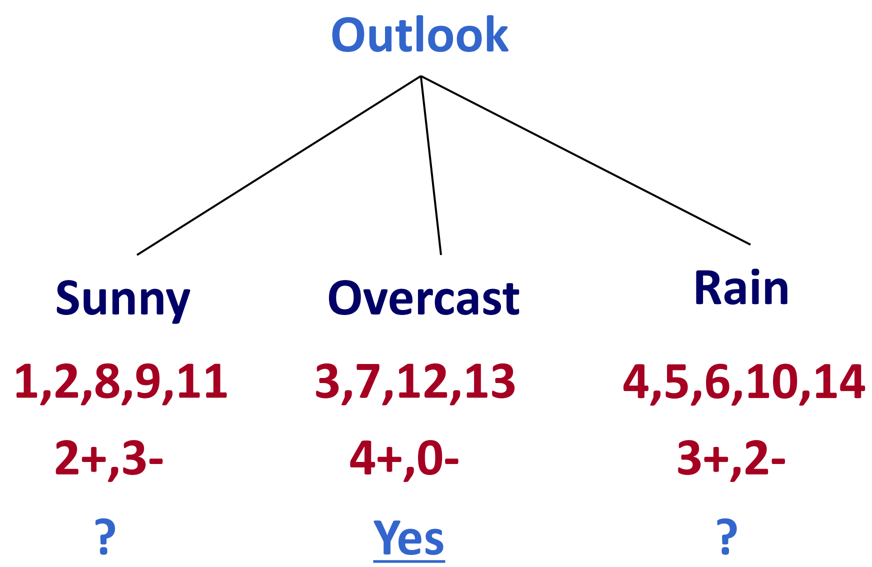
$Gain(S_{sunny}, \text{wind}) = .97-(2/5)\,1 - (3/5)\,.92 = .02$

Outlook

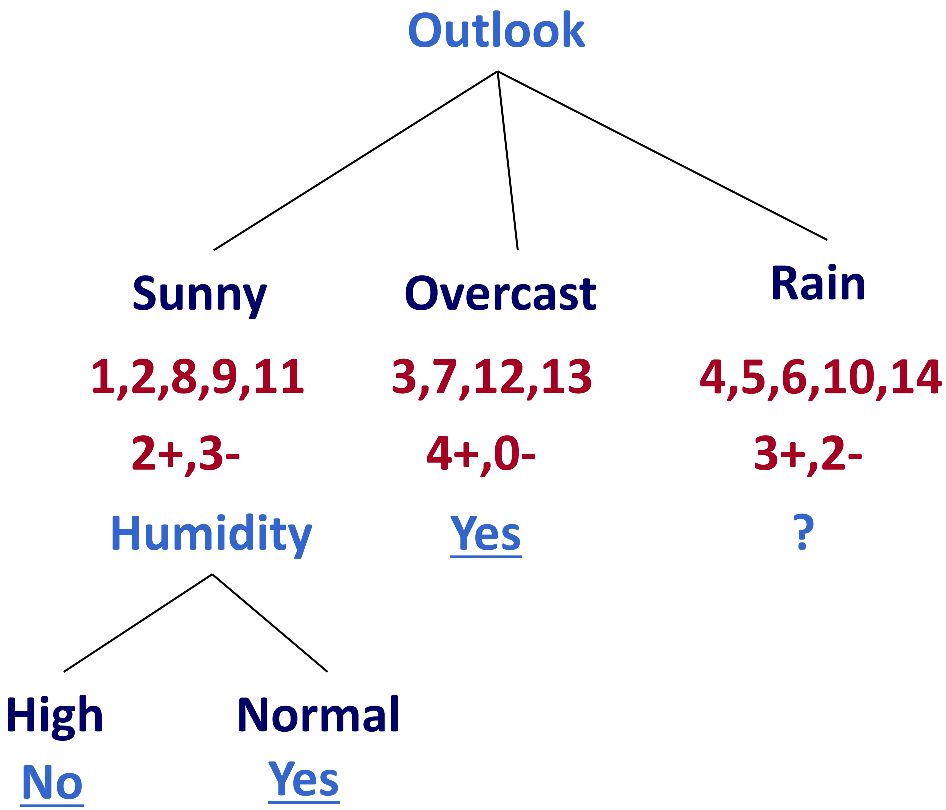
Sunny	Overcast	Rain
1,2,8,9,11	3,7,12,13	4,5,6,10,14
2+,3-	4+,0-	3+,2-
?	<u>Yes</u>	?

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

An Illustrative Example



An Illustrative Example



induceDecisionTree(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:

   $i = \text{argmax}_{\ i}\ \mathit{Gain}(S, X_i)$

3. Add children to S:

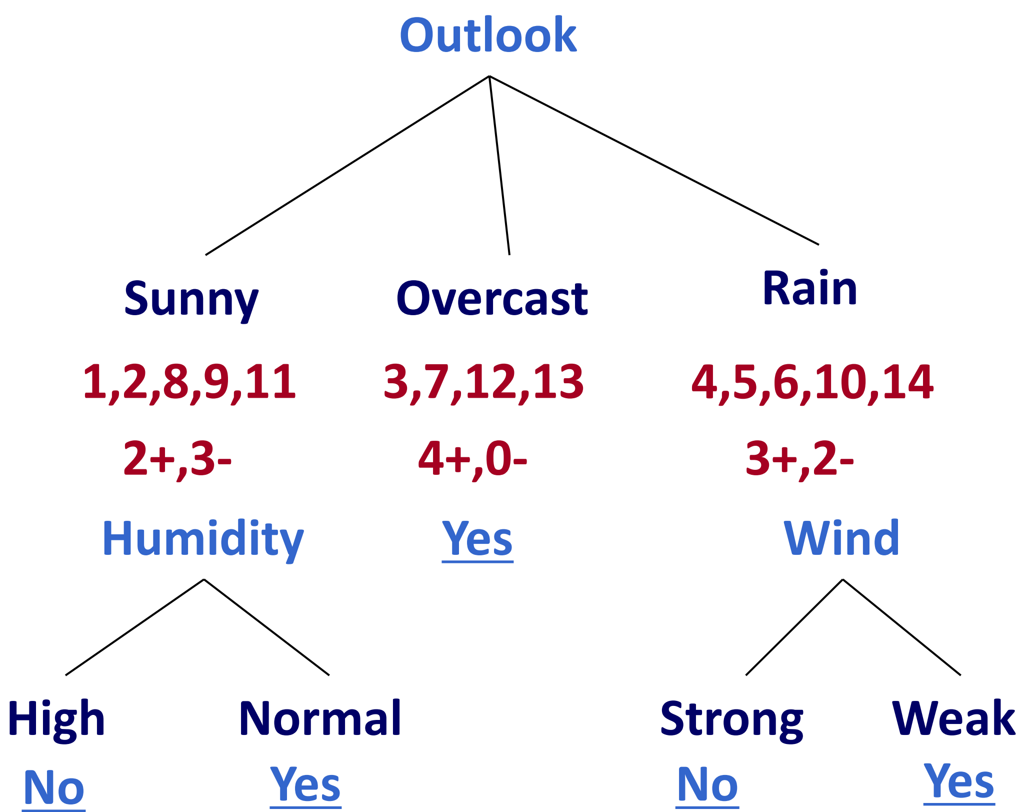
   for *k* **in** Values(X_i):

   $S_k = \{s \in S \mid x_i = k\}$
   addChild(S, S_k)

   induceDecisionTree(S_k)

   return S;

An Illustrative Example



Summary: Learning Decision Trees

1. **Representation**: What are decision trees?
   - A hierarchical data structure that represents data
2. **Algorithm**: Learning decision trees

   The ID3 algorithm: A greedy heuristic
   - If all the examples have the same label, create a leaf with that label
   - Otherwise, find the "most informative" attribute and split the data for different values of that attributes
   - Recurse on the splits