Decision Trees I

Dr. Alex Williams

August 24, 2020

THE UNIVERSITY OF TENNESSEE KNOXVILLE

COSC 425: Introduction to Machine Learning
Fall 2020 (CRN: 44874)

COSC425: Intro to Machine Learning

Course Time: M/W/F @ 2:15-3:05
CRN: 44874, **Term:** Fall 2020

Instructor: Dr. Alex Williams.
Office Hours: Tues/Thurs 2:00-4:00 (**Reserve online!**)

Teaching Assistant: Zhuohang Li (zli96@vols.utk.edu)
Office Hours: By appointment (Book via e-mail.)

Teaching Assistant: Tuhin Das (tdas1@vols.utk.edu)
Office Hours: By appointment (Book via e-mail.)

Overview • Schedule • References • Office Hours

Note: The materials on this webpage are available on Canvas.

Overview

Machine learning is concerned with computer programs that automatically improve their performance through experience. This course covers the theory and practice of machine learning from a variety of perspectives. We cover topics such as clustering, decision trees, neural network learning, statistical learning methods, Bayesian learning methods, dimension reduction, kernel methods, and reinforcement learning. Programming assignments include implementation and hands-on experiments with various learning algorithms.

Schedule

Note: This schedule is subject to change.

Week	Date	Topic	Notes
1	Aug 19	Introduction	
	Aug 21	Defining Machine Learning	**Reading:** Wagstaff.
2	Aug 24	Decision Trees I	
	Aug 26	Decision Trees II	
	Aug 28	The Limits of Learning	

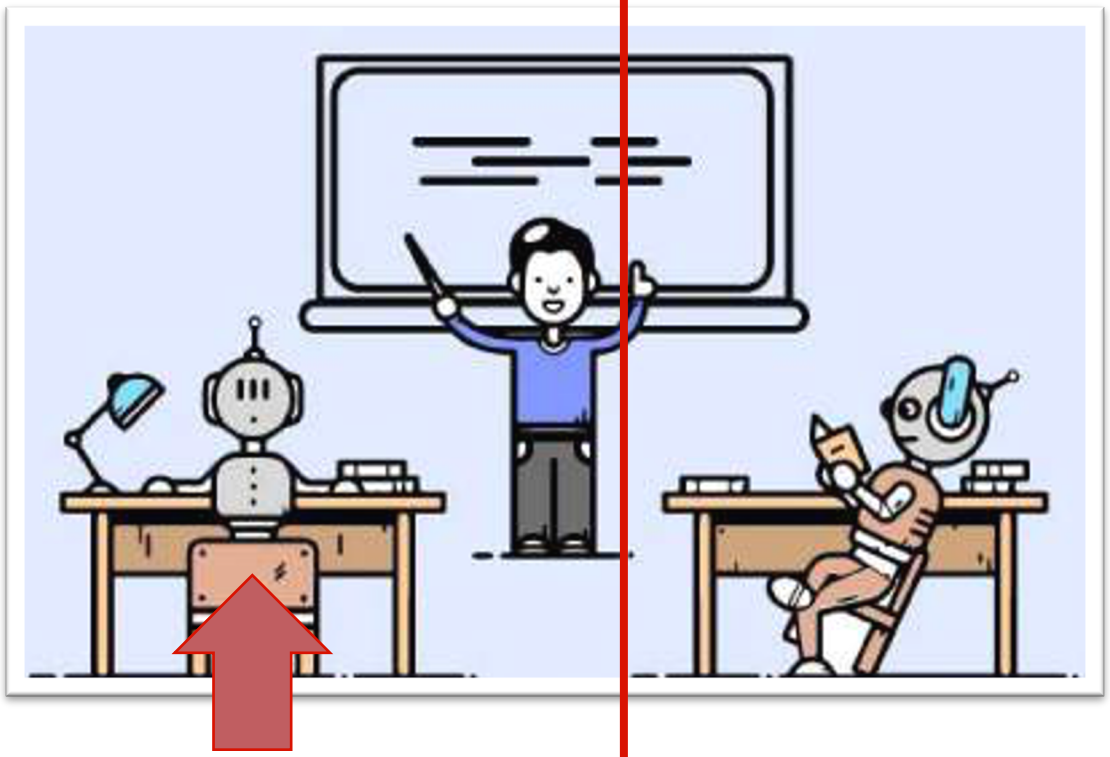
Today's Agenda

We will address:

1. What are decision trees?
2. What functions can we learn with decision trees?

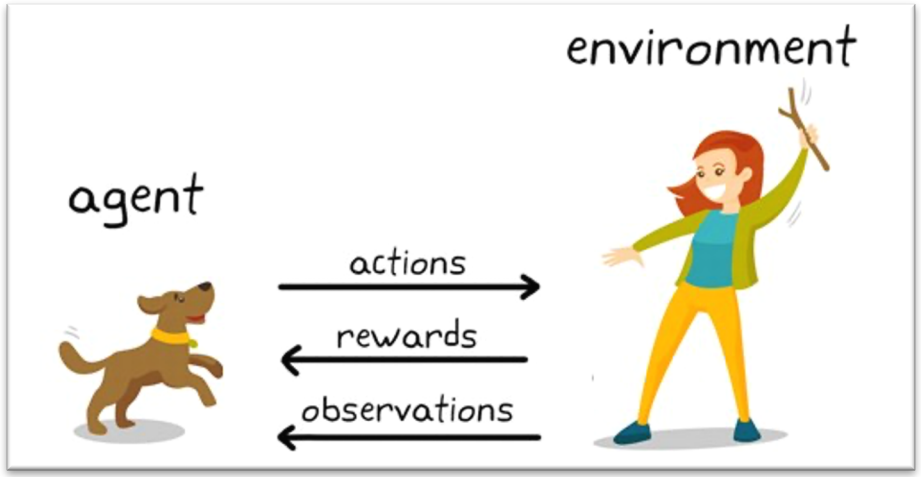
1. What are Decision Trees?

Types of Machine Learning



Supervised Learning

Unsupervised Learning



Reinforcement Learning

Decision Trees: Example Data

Input Variables (Features)

Suppose you have data about students' preferences for courses at UTK.

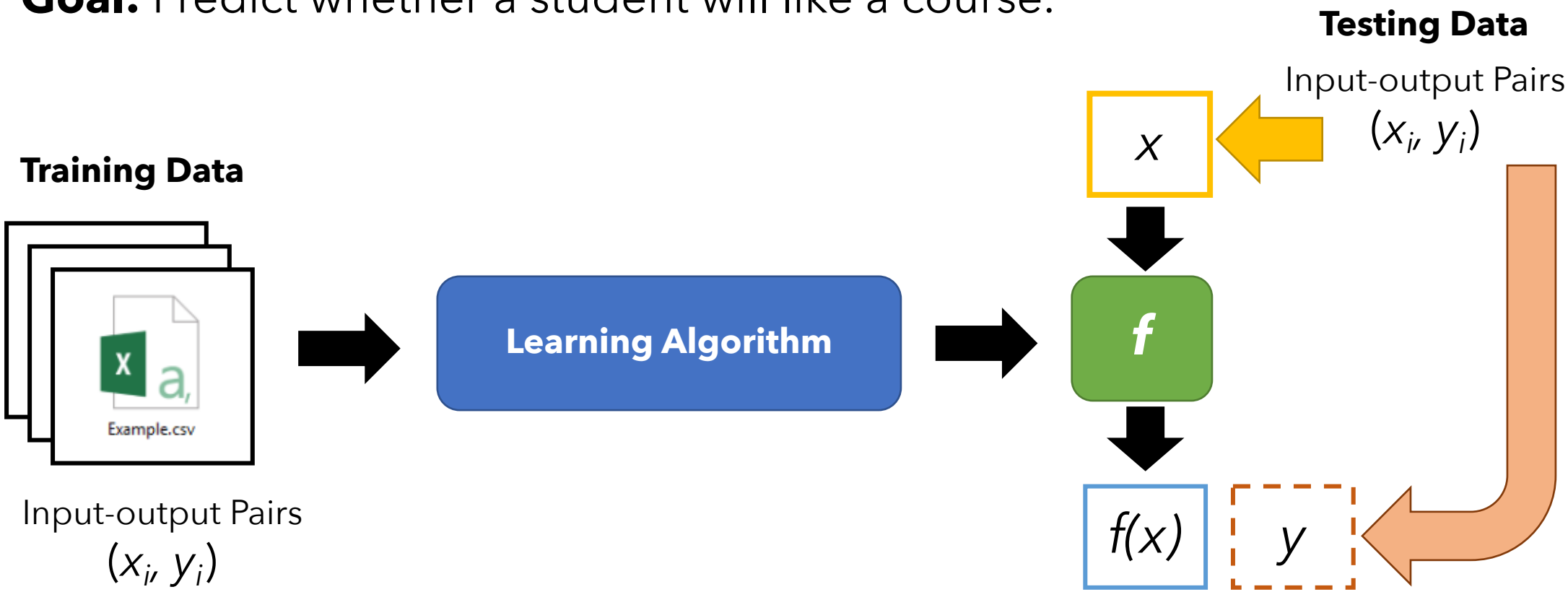
Output Variables (Targets)

student_id	course_type	course_location	difficulty	grade		rating
s1	ML	online	easy	80	…	like
s1	Compilers	face-to-face	easy	87	…	like
s2	Compilers	face-to-face	hard	72	…	dislike
s3	OS	online	hard	79	…	dislike
s3	Algorithms	online	hard	85	…	dislike
s4	ML	online	hard	66	…	like
			…			

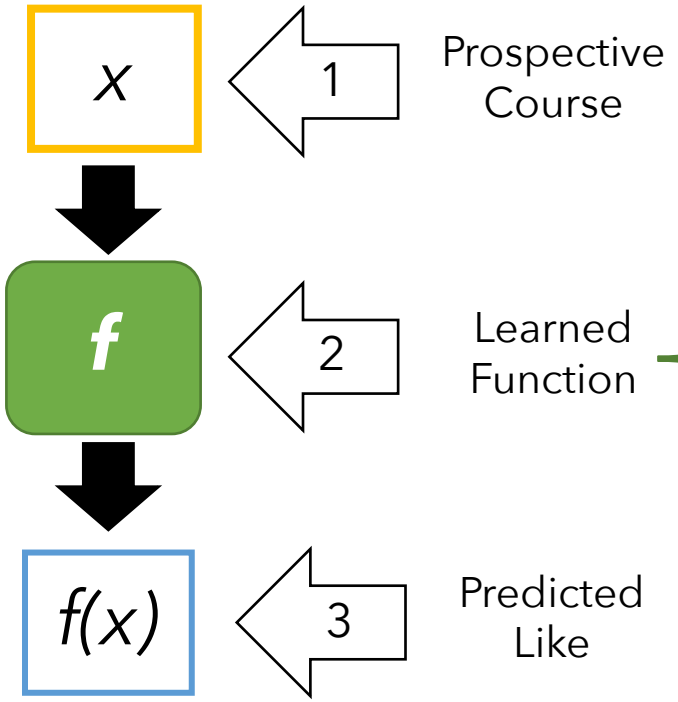
Example / Instance

Dataset (i.e. with Input-Output Pairs)

Goal: Predict whether a student will like a course.



Decision Trees: Questions



Goal: Predict whether a student will like a course.

You: Is the course a Compilers course?

Me: Yes.

You: Is the course online?

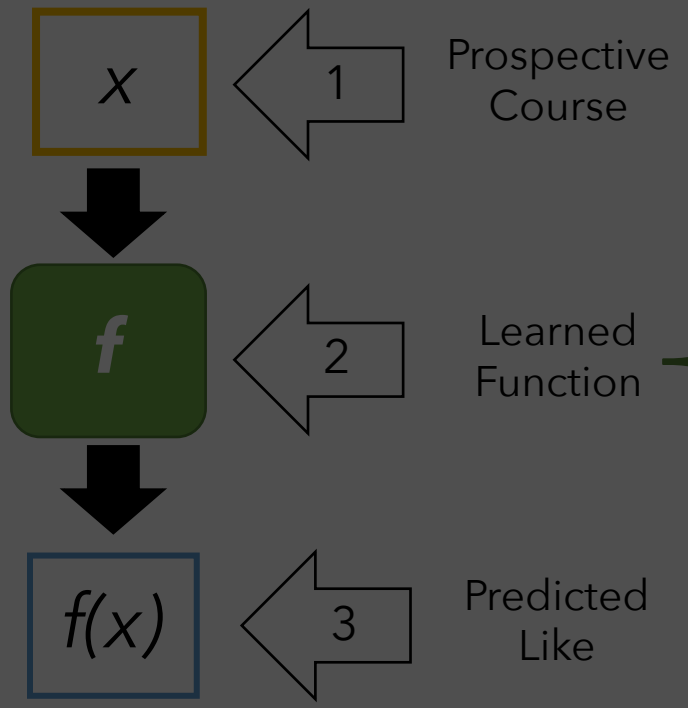
Me: Yes.

You: Were past online courses difficult?

Me: Yes.

You: *I predict the student will not like this course.*

Decision Trees: Questions



Goal: Predict whether a student will like a course.

You: Is the course a Compilers course?

M

You:

M

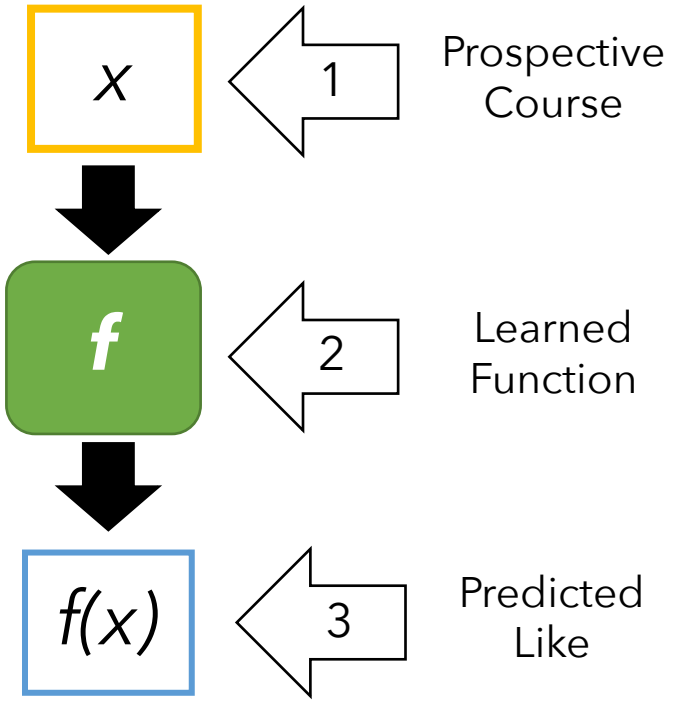
You: rses?

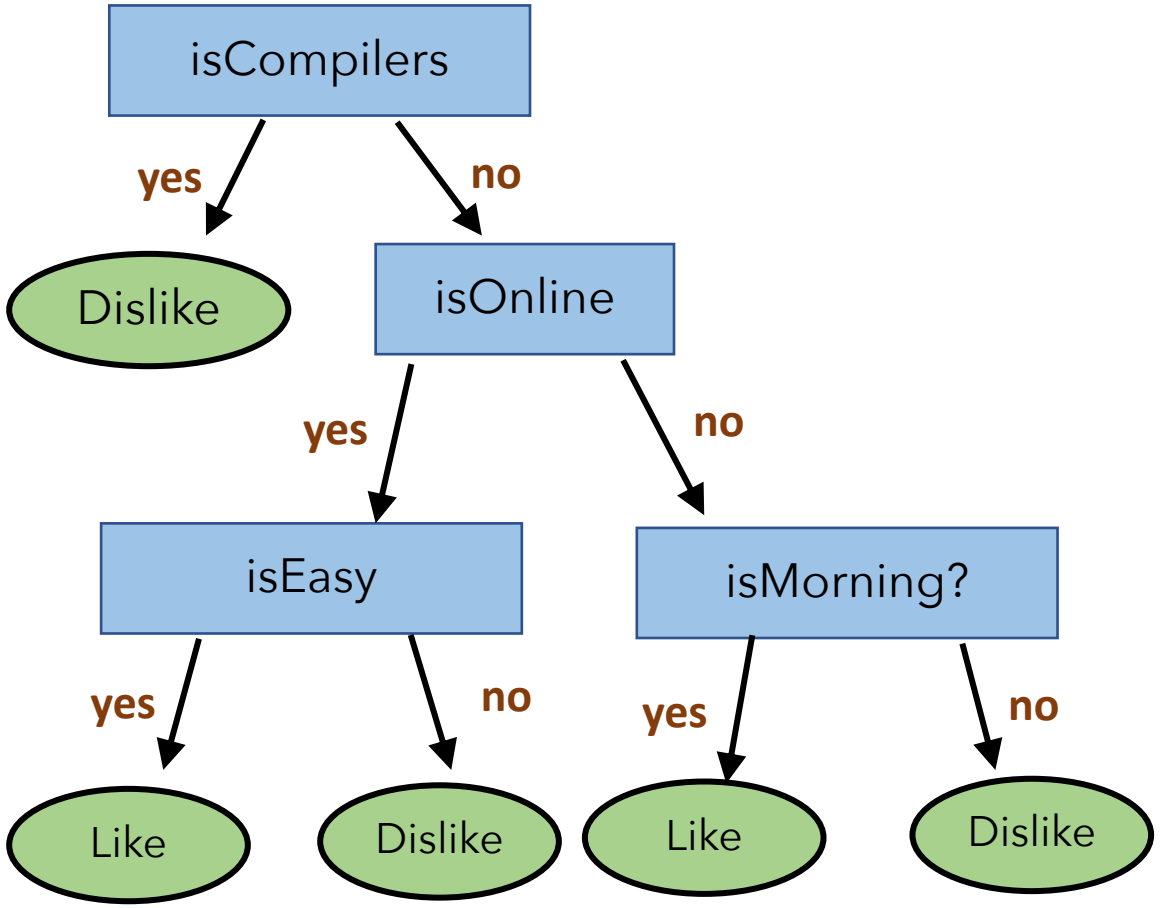
M

You: *I predict the student will not like this course.*

Prediction is about finding questions that matter.

Decision Trees: Questions





From Questions to Learning

Terminology for Decision Trees

instance = a set of feature values

<"Compilers", "online", "easy", 80, …, "like"

question = conditionals constructed based on features

isOnline? **isEasy?** **grade > 80?** **isTaughtByDrWillliams?**

question answer = determined by feature values

yes/no **categorical (e.g. "online", "face-to-face", "hybrid")**

label / target class

"rating"

From Questions to Learning

Learning is concerned with finding the "best" tree for the data.

We could enumerate all possible trees and evaluate each tree.

… Okay. So, how many trees is that? **Answer:** Too many!

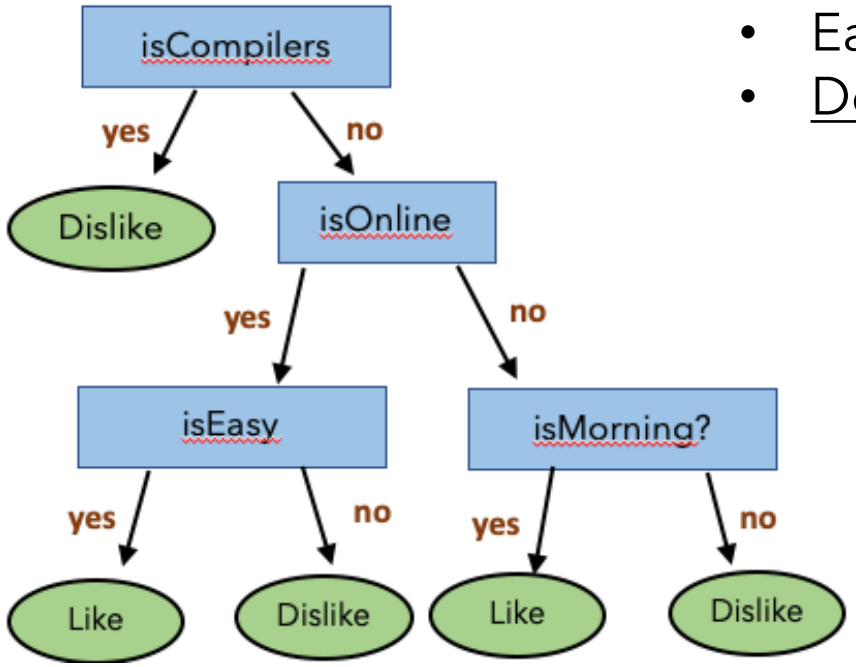
<"Compilers", "online", "easy", 80, …, "like"> → **Finding Optimal Tree is NP-Hard**

(See Hyafil and Rivest 1976.)

Thus: We greedily ask "If I could ask one question, what is it?"

Alternative framing: "What is the one question that would be most helpful in estimating whether a student will enjoy a particular course?"

From Questions to Learning



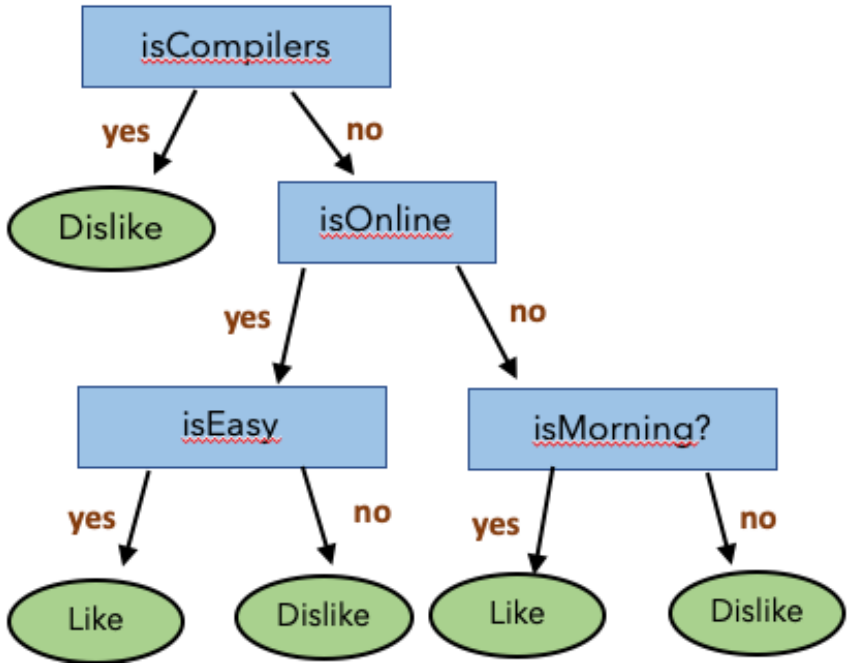
Decision Trees Split Your Data

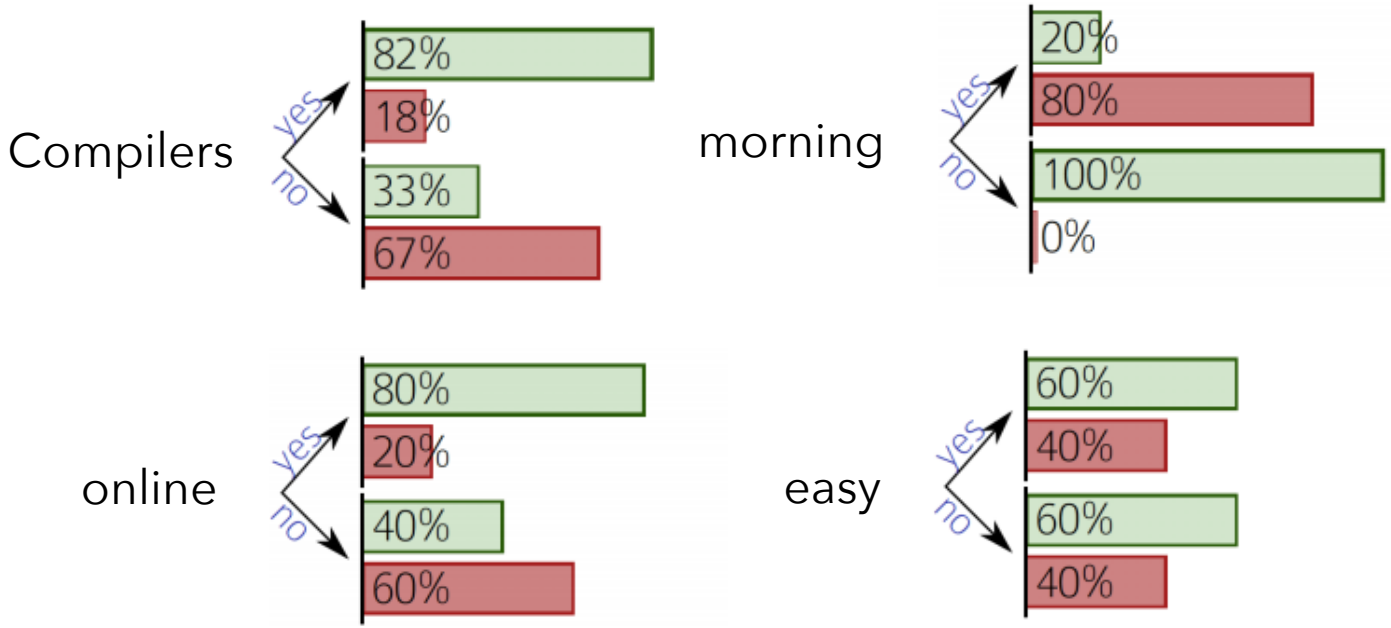
- Each node represents a question that splits your data.
- Decision tree learning = choosing what internal nodes should be.

Questions are Conditionals

- Grade > 80
- Grade in [80-90]
- Location is {“online”, “hybrid”, “face-to-face”}
- Teacher is DR_WILLIAMS
- MLgrade * 2 + COMPILERgrade * 3

From Questions to Learning

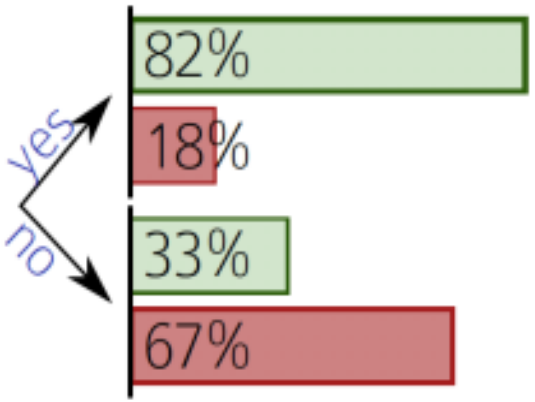




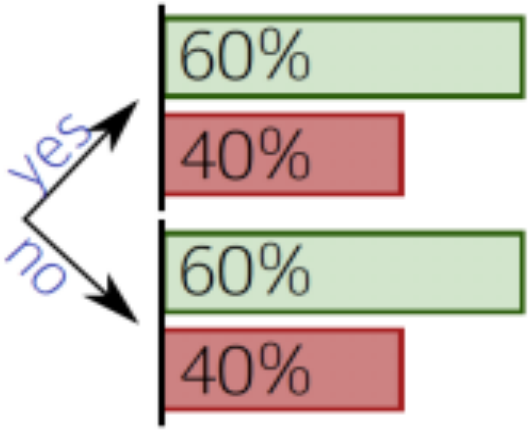
Distribution of Like/Dislike labels for each question.

From Questions to Learning

Compilers



easy

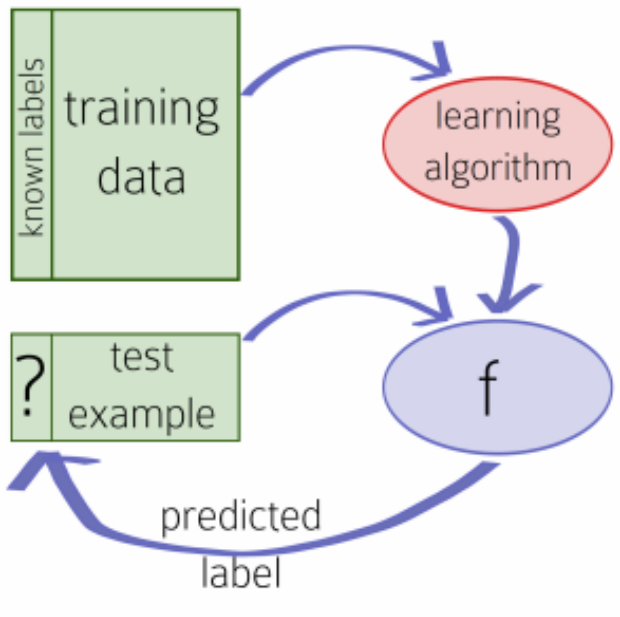


Informative

Uninformative

2. What Functions Can We Learn?

Supervised Learning: Theory



[D] Figure1.1
(Daumé, pg. 9)

Problem Setting:

- Set of possible instances: X
- Unknown target function: $f : X \rightarrow Y$
- Set of function hypotheses

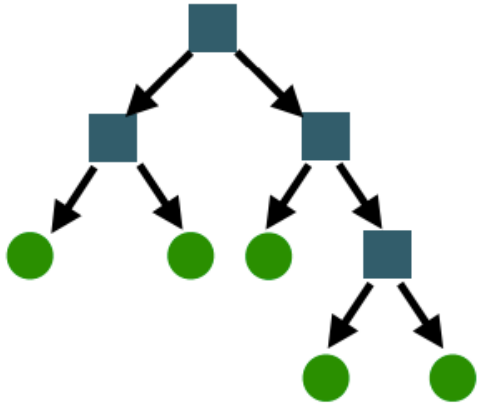
$$H = \{h \mid h : X \rightarrow Y\}$$

The Learning Algorithm:

- *Input*: training examples $\langle x_i, y_i \rangle$
- *Output*: Hypothesis $h \in H$ that best approximates the target function f

The set of all hypotheses that can be "spat out" by a learning algorithm is called the **hypothesis space**.

Supervised Learning: Theory



Problem Setting:

- Set of possible instances: X

  Each instance is a feature vector.

- Unknown target function: $f : X \rightarrow Y$

  y = 1 if a student likes the course; otherwise, y = 0

- Set of function hypotheses

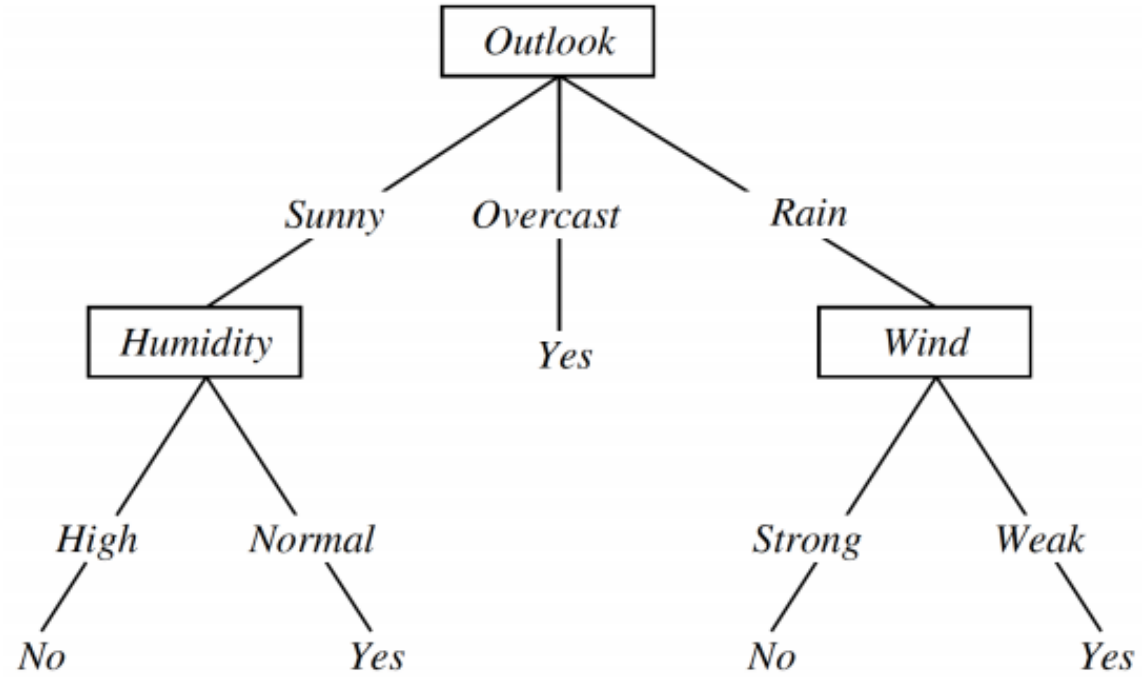
$$H = \{h \mid h : X \rightarrow Y\}$$

Each hypothesis is a decision tree!

The Learning Algorithm:

- *Input*: training examples $\langle x_i, y_i \rangle$
- *Output*: Hypothesis $h \in H$ that best approximates the target function f

Trees as Functions: Boolean Logic

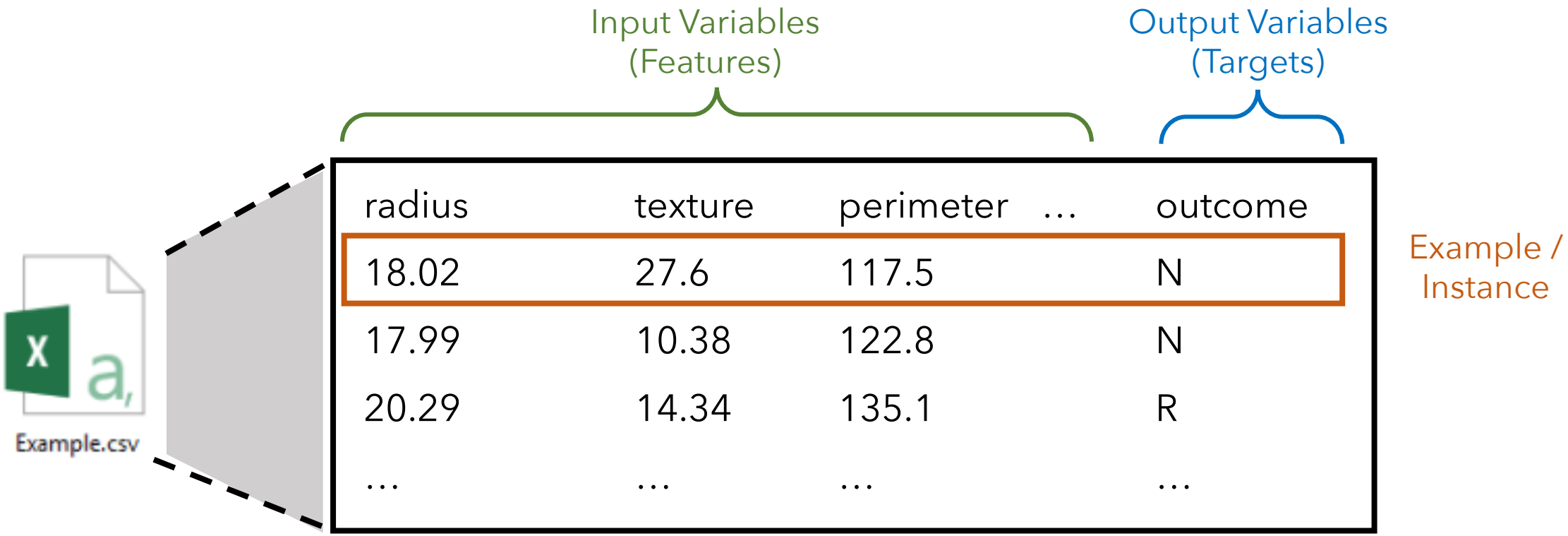


Translate the Tree to Boolean Logic

$$(\text{Outlook} = \text{Sunny} \wedge \text{Humidity} = \text{Normal})$$
$$\vee \quad (\text{Outlook} = \text{Overcast})$$
$$\vee \quad (\text{Outlook} = \text{Rain} \wedge \text{Wind} = \text{Weak})$$

Example: Weather Prediction

Example: Cancer Recurrence Prediction



Output Variables: **N** = No Recurrence; **R** = Recurrence

Example: Cancer Recurrence Prediction

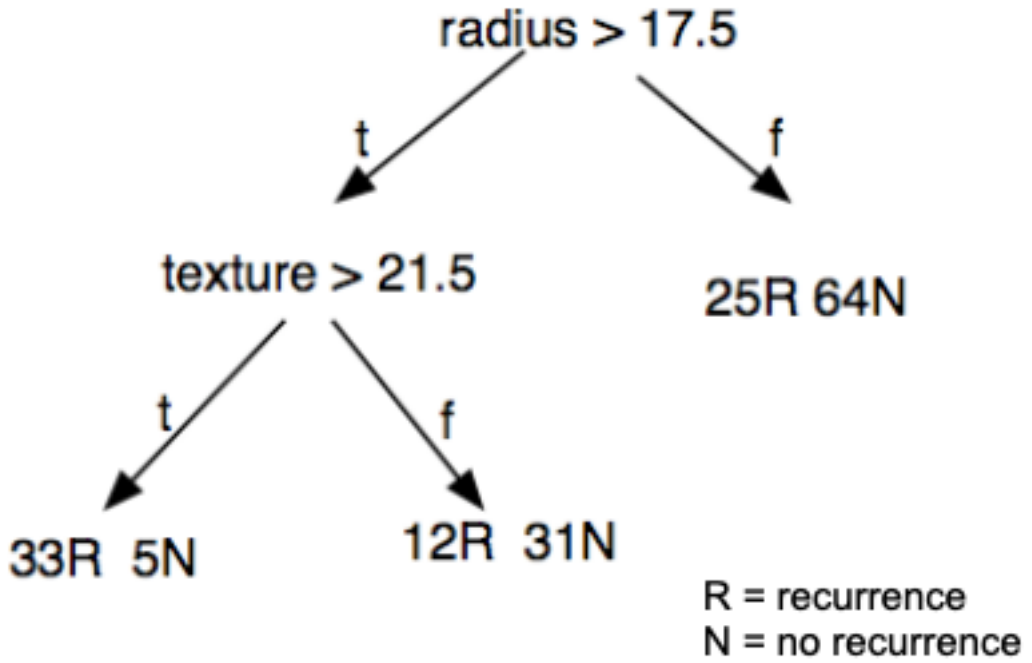
What does a node present?

A partitioning of the input space.

Internal Nodes: A test or question.

- *Discrete features*: Branch on all values
- *Real Features*: Branch on threshold value

Leaf Nodes: Include instances that satisfy the tests along the branch.



Remember the Following:

- Each instance maps to a particular leaf.
- Each leaf typically contains more than one example.

Example: Cancer Recurrence Prediction

Input Variables (Features)

Output Variables (Targets)

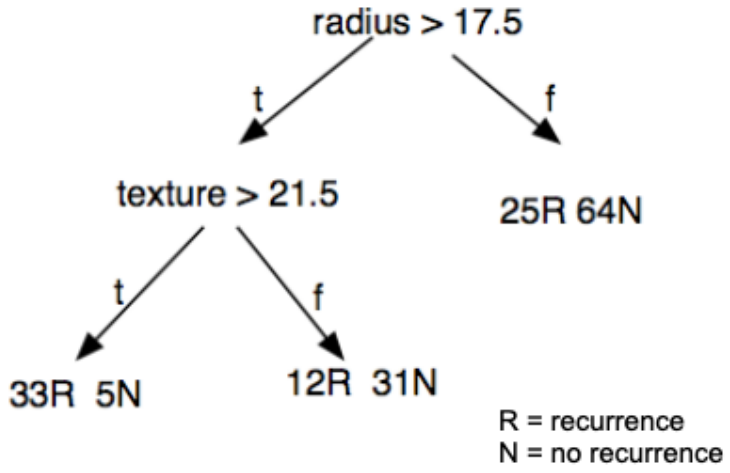
radius	texture	perimeter	…	outcome
18.02	27.6	117.5		N
17.99	10.38	122.8		N
20.29	14.34	135.1		R
…	…	…		…

Example / Instance

Output Variables: **N** = No Recurrence; **R** = Recurrence

Example: Cancer Recurrence Prediction

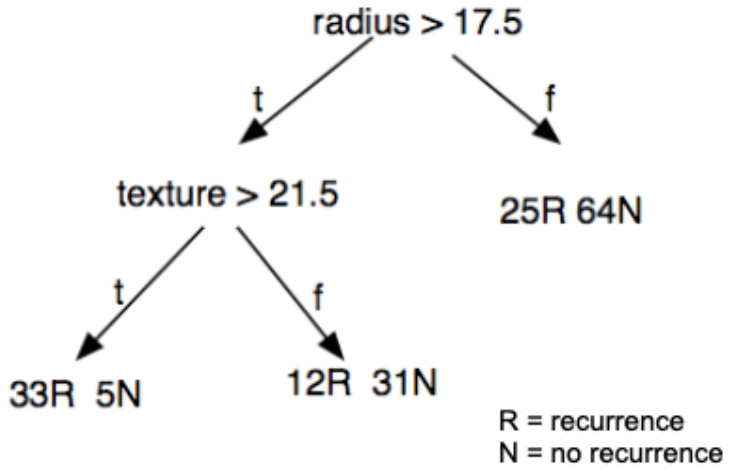
Conversion: Decision trees translate to sets of if-then rules.



IF	THEN most likely class is
radius>17.5 AND texture>21.5	R
radius>17.5 AND texture≤21.5	N
radius≤17.5	N

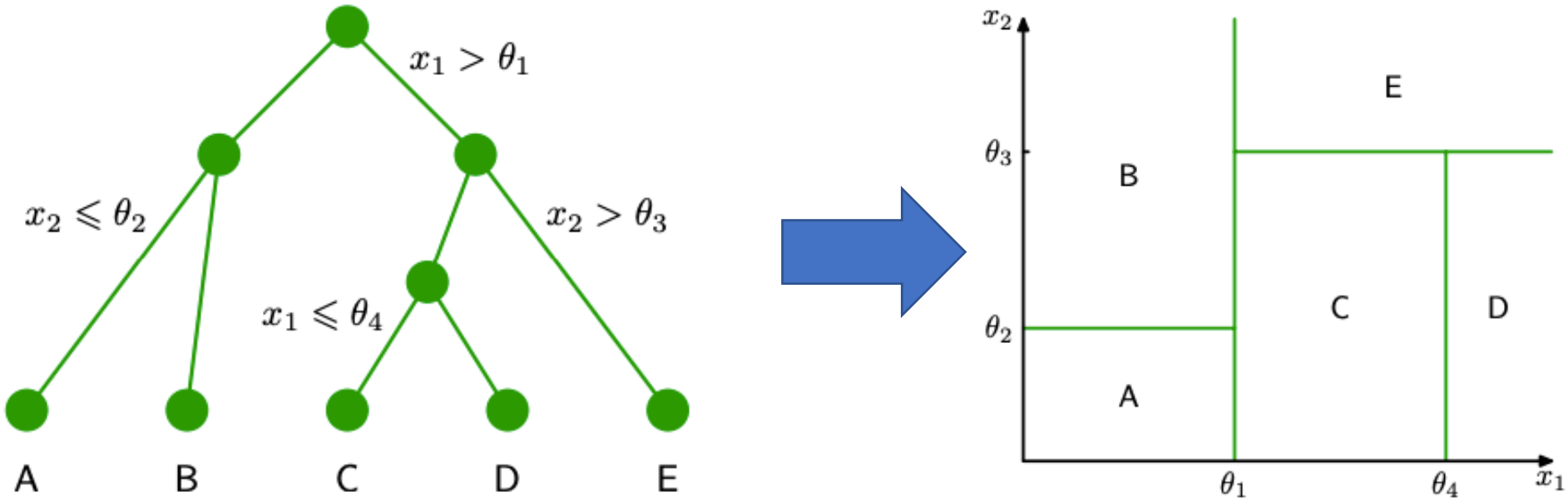
Example: Cancer Recurrence Prediction

Conversion: Decision trees can represent probability of recurrence.



IF	THEN P(R) is
radius>17.5 AND texture>21.5	$\frac{33}{33+5}$
radius>17.5 AND texture≤21.5	$\frac{12}{12+31}$
radius≤17.5	$\frac{25}{25+64}$

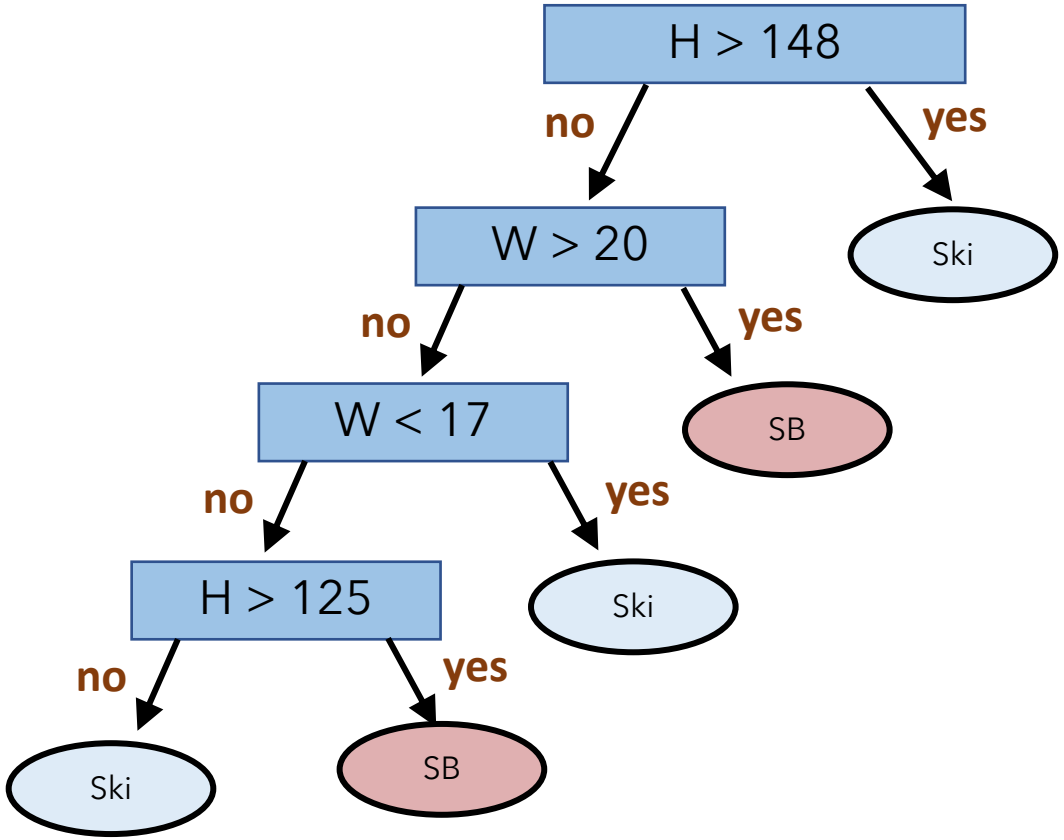
Decision Trees: Interpretation

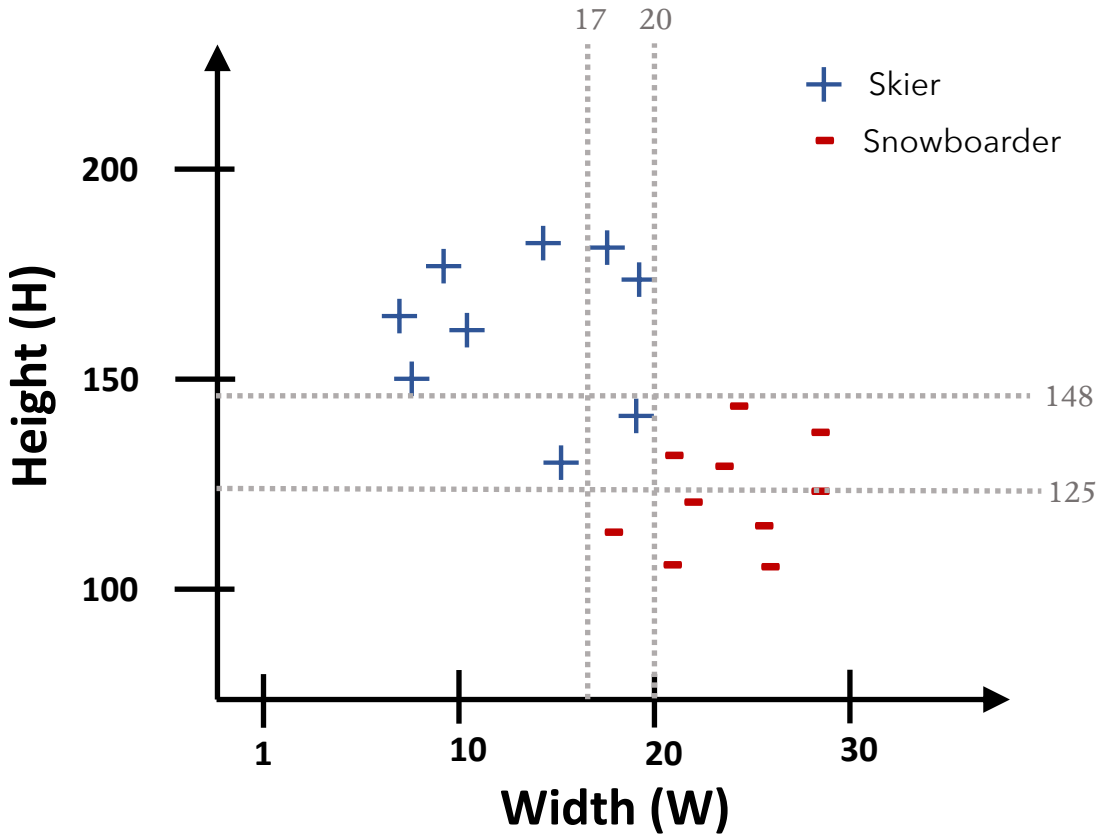


Important: Decision trees form boundaries in your data.

Decision Trees: Interpretation

Predict a person's interest in skiing or snowboarding.





Decision Trees: Interpretation

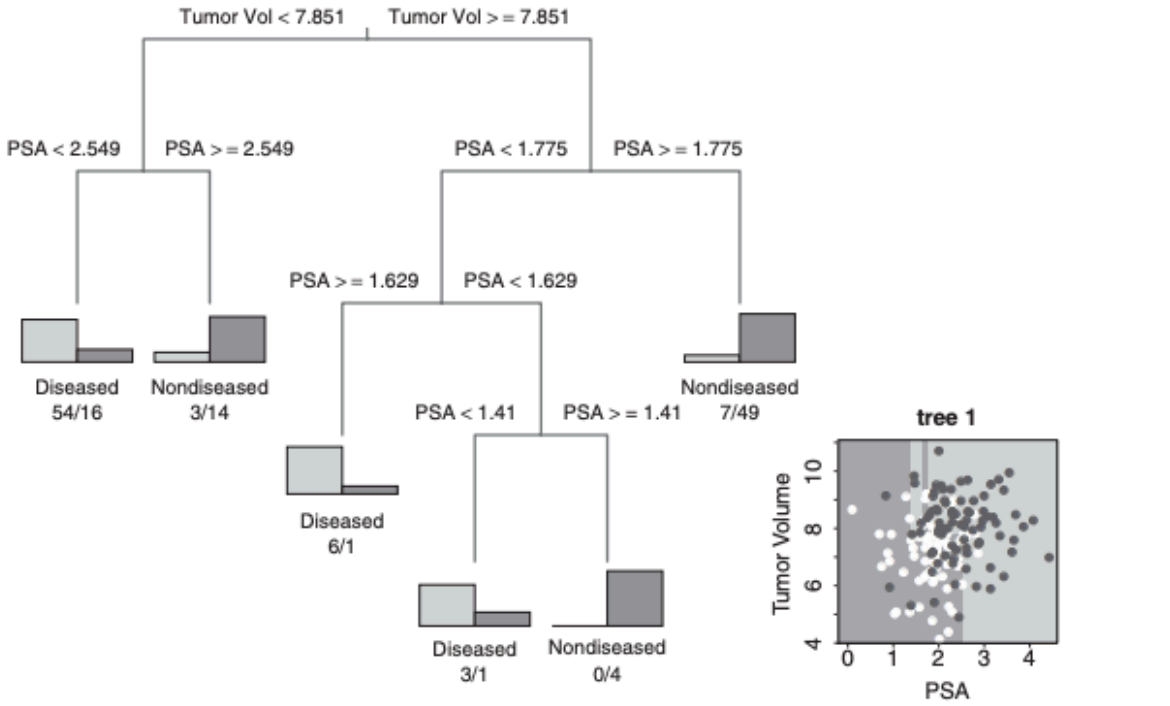
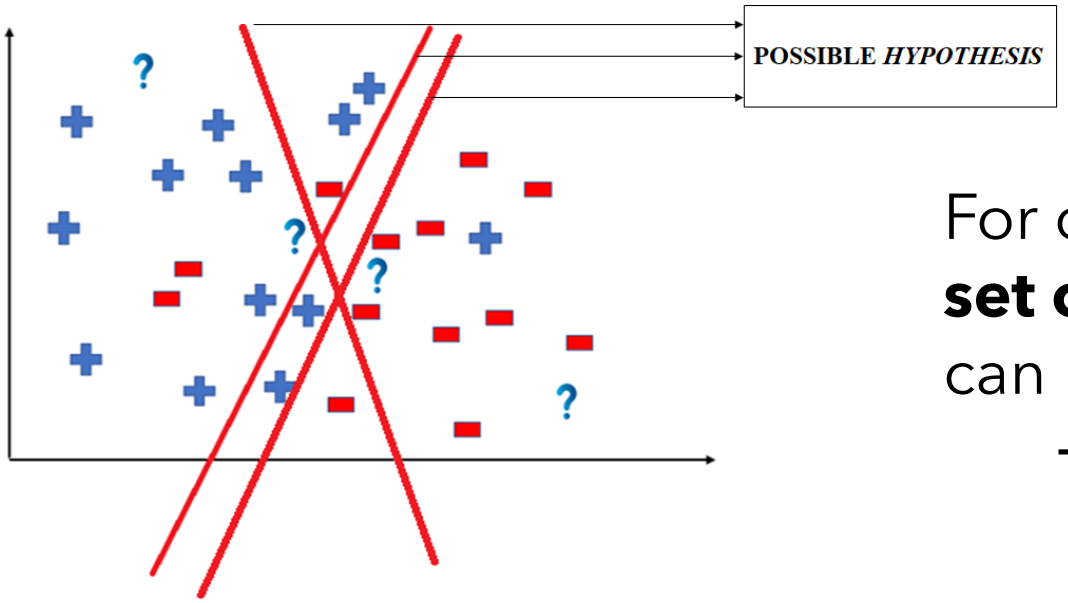


Figure 1 Decision tree (left-hand side) and decision boundary (right-hand side) for prostate cancer data with prostate-specific antigen (PSA) and tumor volume as independent variables (both transformed on the log scale)

See Ishwaran H. and Rao J.S. (2009)

Hypothesis Space

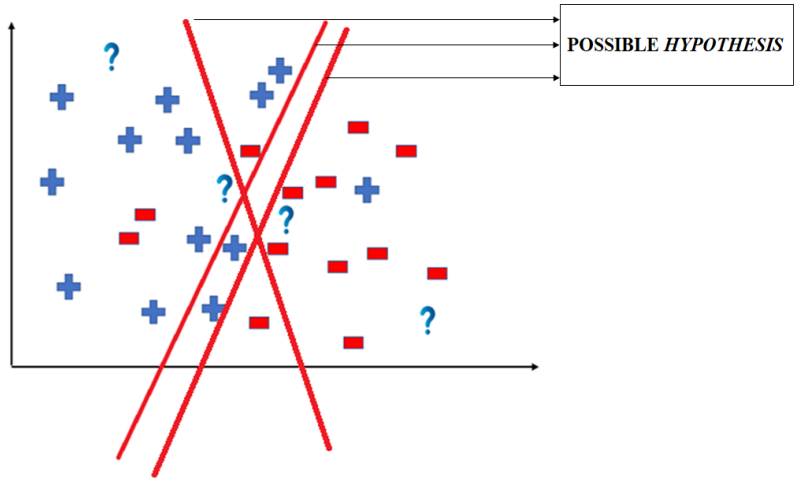


For decision trees, the hypothesis space is **the set of all possible finite discrete functions** that can be learned based on the data.

→ $f(x) = \{ \text{category}1, \text{category}2, \ldots, \text{category}N \}$

Every finite discrete function can be represented by some decision tree.
(… Hence the need to be greedy!)

Hypothesis Space



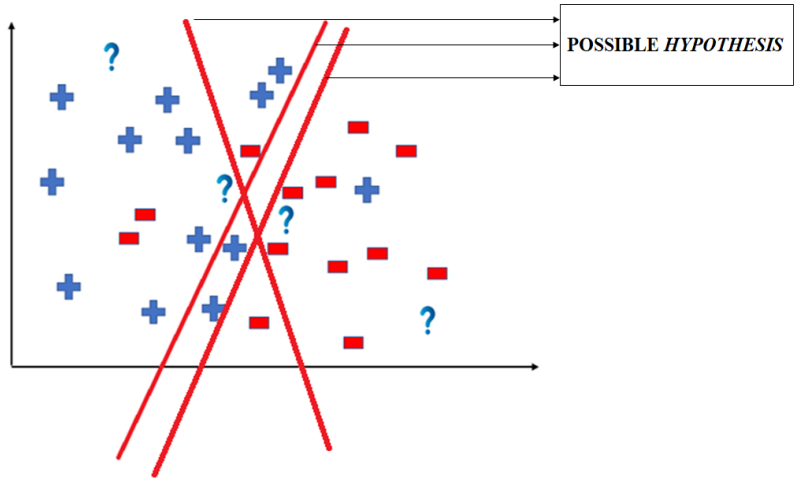
Note: Encoding Challenges

- Some functions demand exponentially large decision trees to represent.

Boolean functions can be fully expressed in decision trees.

- Each entry in a truth table can be one path. (Inefficient!)
- Most Boolean functions can be encoded more compactly.

Decision Boundaries



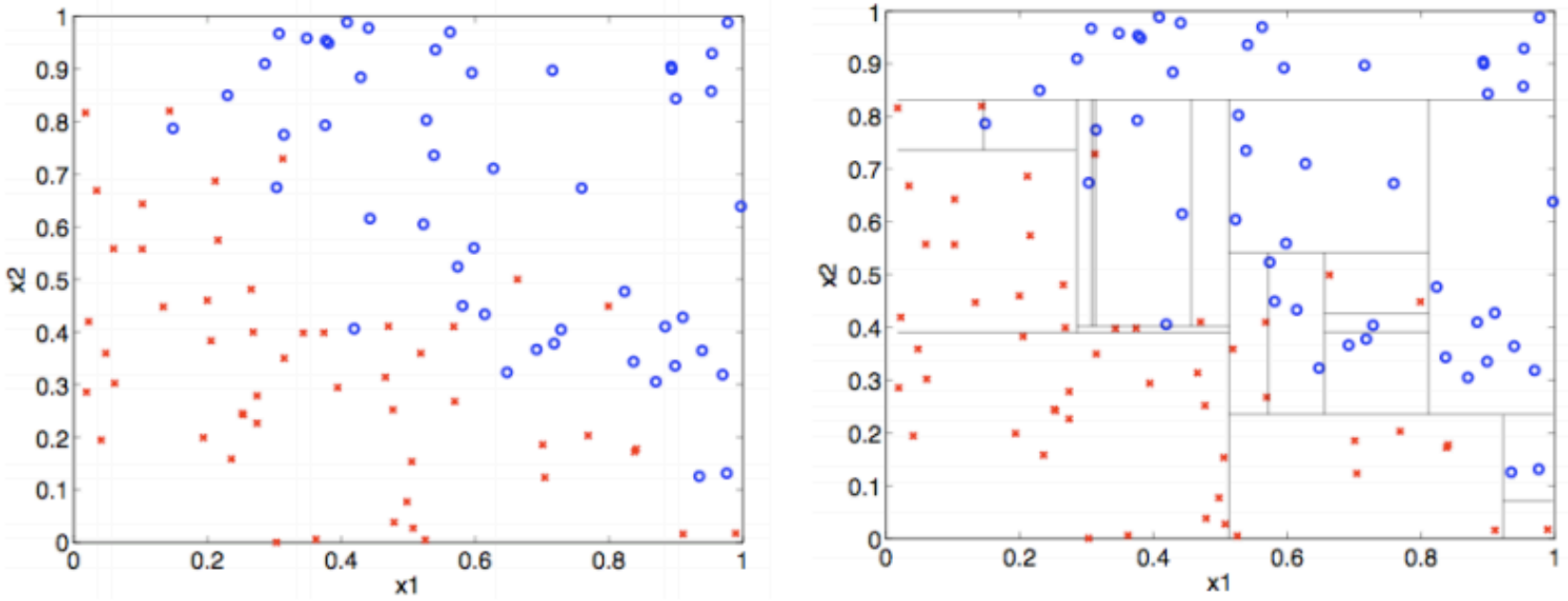
Note: Encoding Challenges

- Some functions demand exponentially large decision trees to represent.

Boolean functions can be fully expressed in decision trees.

- Each entry in a truth table can be one path. (Inefficient!)
- Most Boolean functions can be encoded more compactly.

Decision Boundaries for Real-Valued Features



Use Real-Valued Features with "Nice" Bounds.

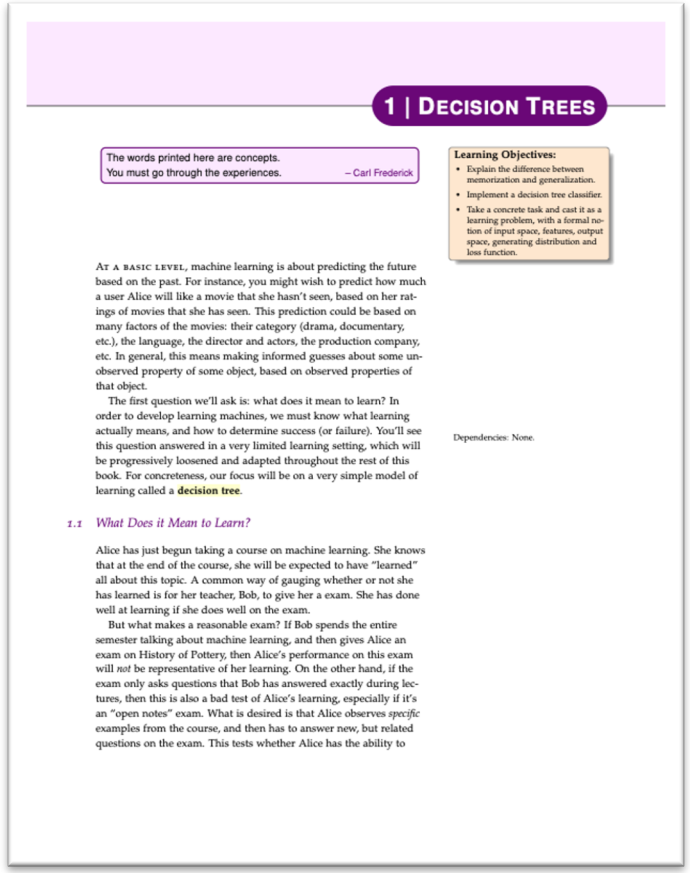
- Best used when labels occupy "axis-orthogonal" regions of input space.

Today's Agenda

We have addressed:

1. What are decision trees?
2. What functions can we learn with decision trees?

Reading



1 | DECISION TREES

The words printed here are concepts.
You must go through the experiences. – Carl Frederick

Learning Objectives:
- Explain the difference between memorization and generalization.
- Implement a decision tree classifier.
- Take a concrete task and cast it as a learning problem, with a formal notion of input space, features, output space, generating distribution and loss function.

Dependencies: None.

AT A BASIC LEVEL, machine learning is about predicting the future based on the past. For instance, you might wish to predict how much a user Alice will like a movie that she hasn't seen, based on her ratings of movies that she has seen. This prediction could be based on many factors of the movies: their category (drama, documentary, etc.), the language, the director and actors, the production company, etc. In general, this means making informed guesses about some unobserved property of some object, based on observed properties of that object.

The first question we'll ask is: what does it mean to learn? In order to develop learning machines, we must know what learning actually means, and how to determine success (or failure). You'll see this question answered in a very limited learning setting, which will be progressively loosened and adapted throughout the rest of this book. For concreteness, our focus will be on a very simple model of learning called a **decision tree**.

1.1 What Does it Mean to Learn?

Alice has just begun taking a course on machine learning. She knows that at the end of the course, she will be expected to have "learned" all about this topic. A common way of gauging whether or not she has learned is for her teacher, Bob, to give her a exam. She has done well at learning if she does well on the exam.

But what makes a reasonable exam? If Bob spends the entire semester talking about machine learning, and then gives Alice an exam on History of Pottery, then Alice's performance on this exam will *not* be representative of her learning. On the other hand, if the exam only asks questions that Bob has answered exactly during lectures, then this is also a bad test of Alice's learning, especially if it's an "open notes" exam. What is desired is that Alice observes *specific* examples from the course, and then has to answer new, but related questions on the exam. This tests whether Alice has the ability to

Daume. Chapter 1

Next Time

We will address:

1. How do you train and test decision trees?
2. How can decision trees generalize?
3. What is the inductive bias of decision trees?