Warming up …

- **Which of these is a ML application?**
  - Face recognition
  - Finding the shortest path between two addresses
  - Finding the fastest path between two addresses
  - Searching for videos and images
- **Is it always the case that training data has labels?**
- **Which type of ML is the following task?**
  - Predicting the price of new house
  - A machine playing football
  - Grouping similar news together
  - Predicting the genre of a book
  - Predicting the age of a person
  - Predicting the gender of a person

جامعة قطر
QATAR UNIVERSITY

CMPS 460 – Spring 2022

MACHINE LEARNING

Tamer Elsayed

Image hosted by: WittySparks.com | Image source: Pixabay.com

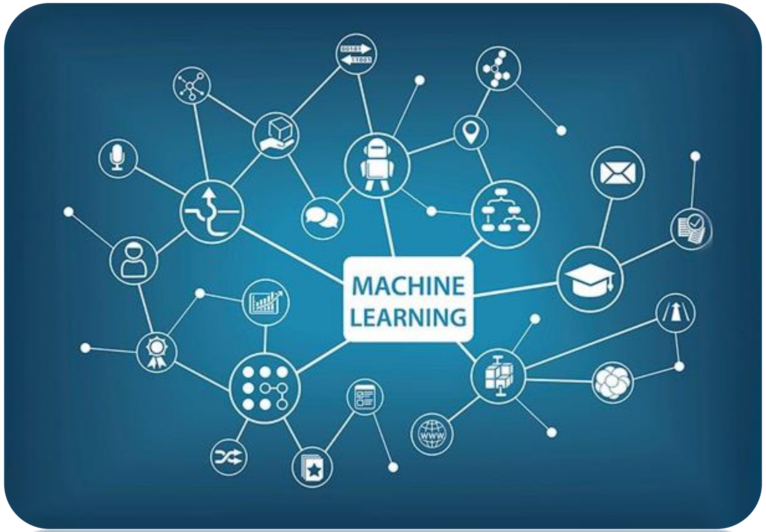
1.c

Decision Trees

Chapter 1

Why Decision Trees (DTs) First?

- A classic and natural model of learning.
- Easy to understand.
- Closely related to the fundamental computer science notion of "divide and conquer."



What is a Decision Tree?

Example: Play Tennis?

- Given some attributes of the current weather
- Should I play tennis today or not?

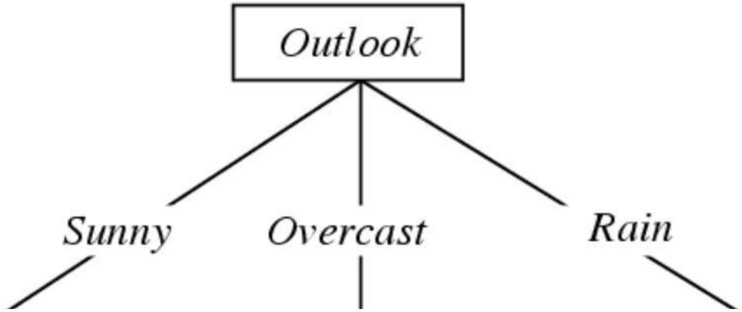
Weather attributes

- **Outlook**: sunny, overcast, rain
- **Temperature**: hot, mild, cool
- **Humidity**: high, normal
- **Wind**: weak, strong

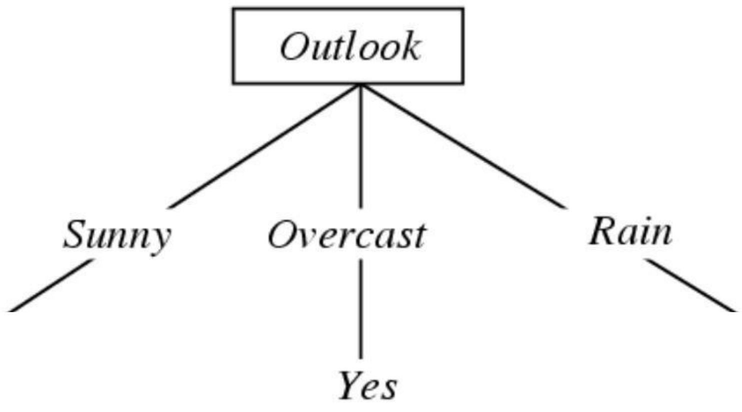
An example training set

Day	Outlook	Temperature	Humidity	Wind	PlayTennis?
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

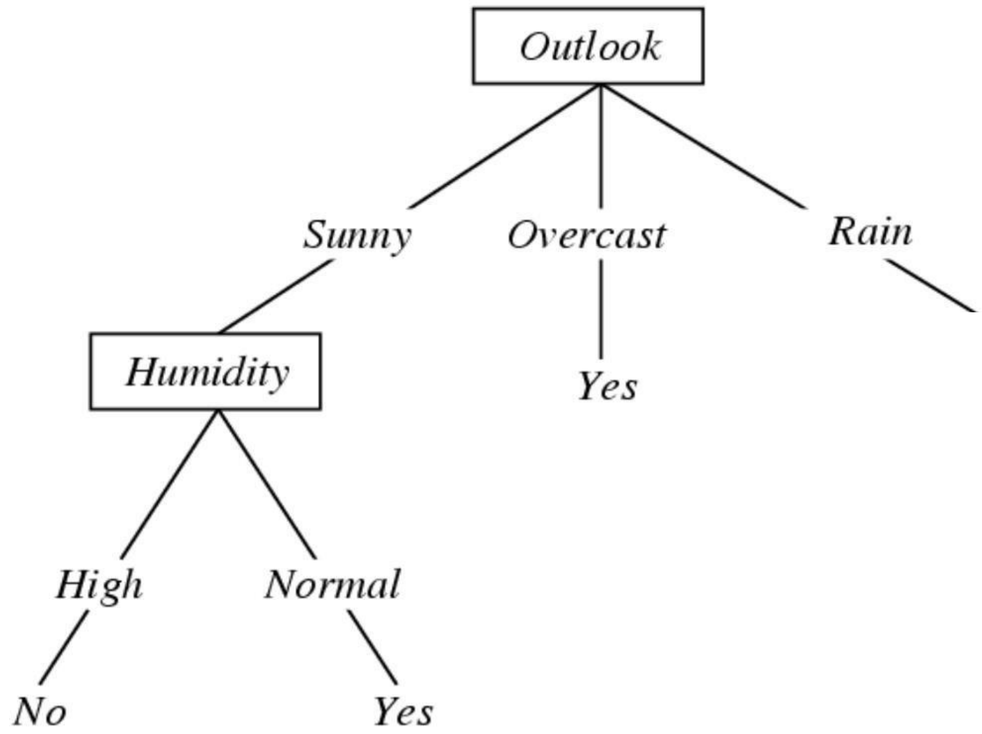
Classify a new day?



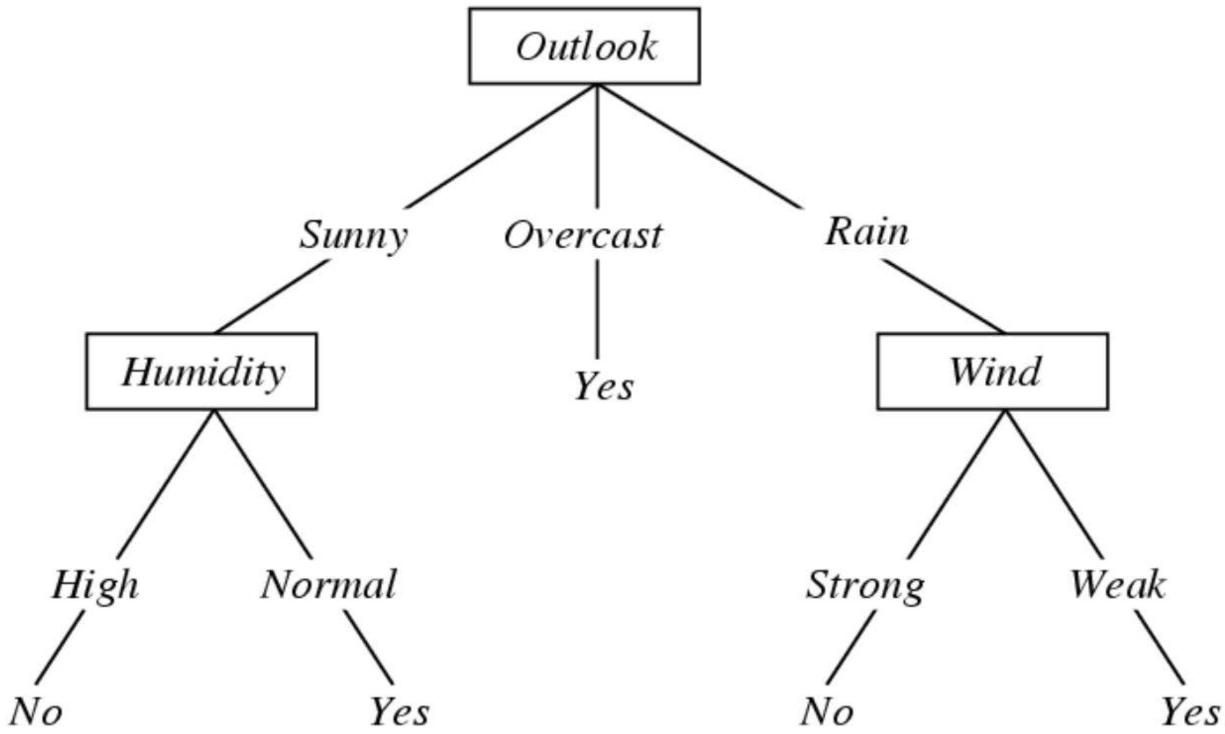
Classify a new day?



Classify a new day?



Classify a new day?

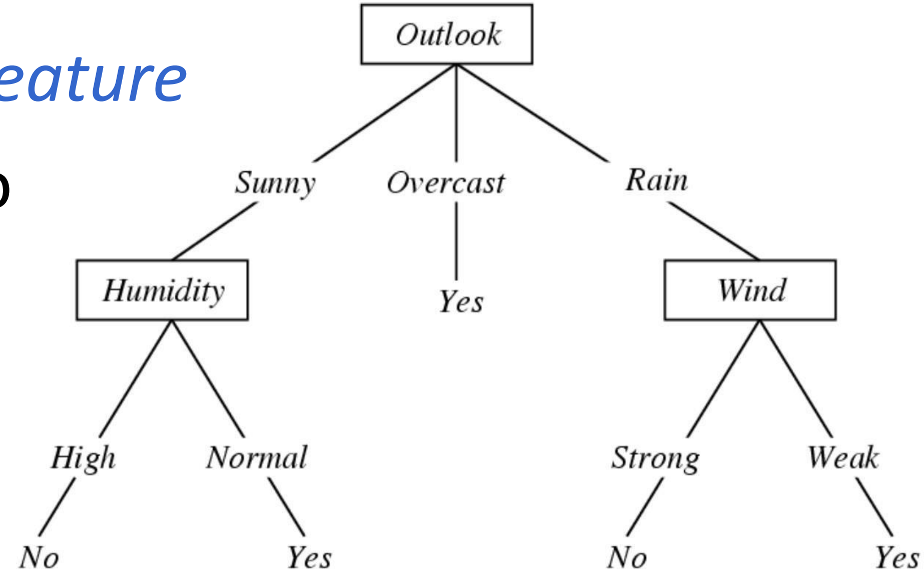


A possible "decision tree"

Decision Trees

Representation

- Each internal node tests a *feature*
- Each branch corresponds to a *feature value*
- Each leaf node assigns a *classification*



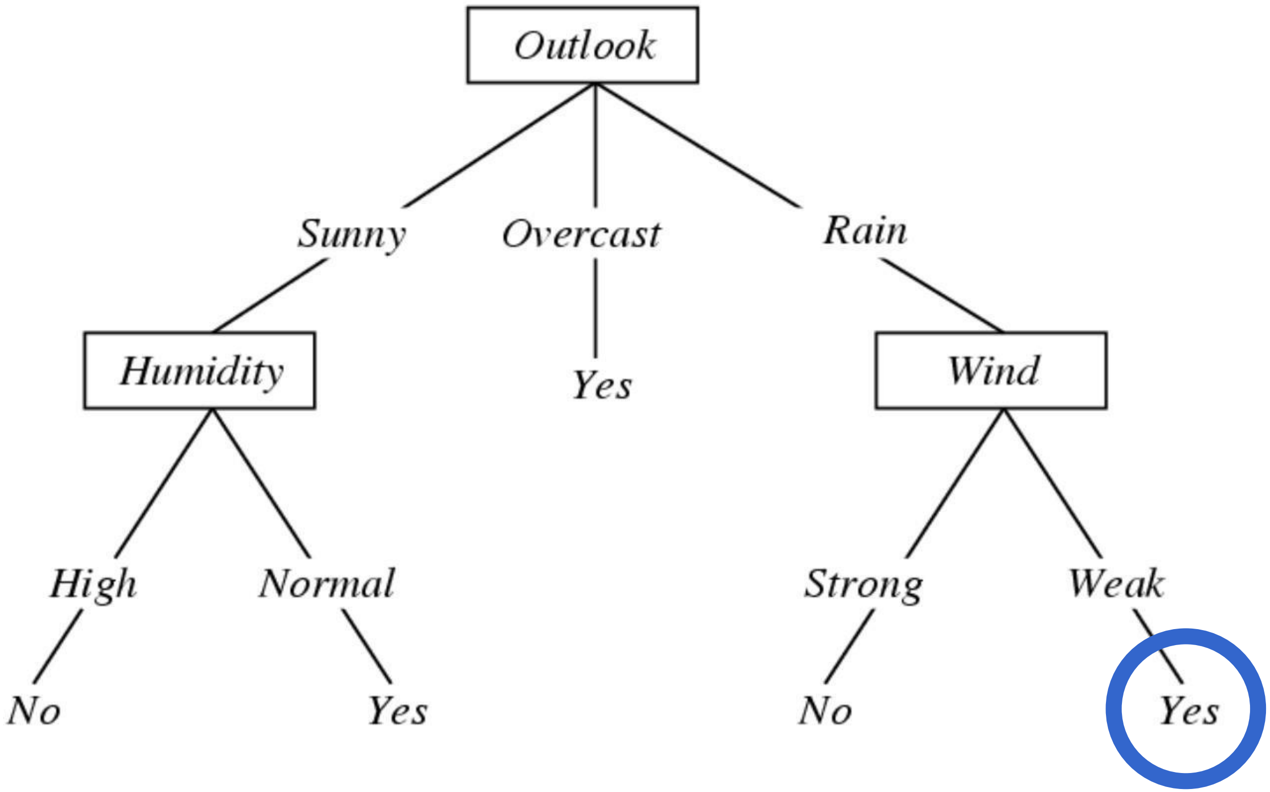
- Decision trees represent functions that map examples in X to classes in Y

  f: *<Outlook, Temp, Humidity, Wind>* → *PlayTennis*?

Predict

<Outlook, Temp, Humidity, Wind>

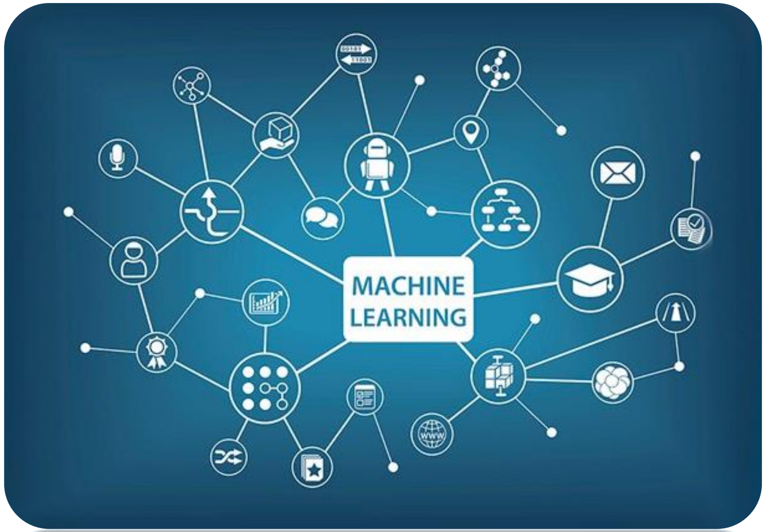
<Rain, Hot, High, Weak>



Exercise: Boolean Functions as DTs

How would you represent the following Boolean functions with decision trees?

- $A \cap B$
- $A \cup B$
- $A \oplus B$
- $A \cap B \cup (C \cap \neg D)$

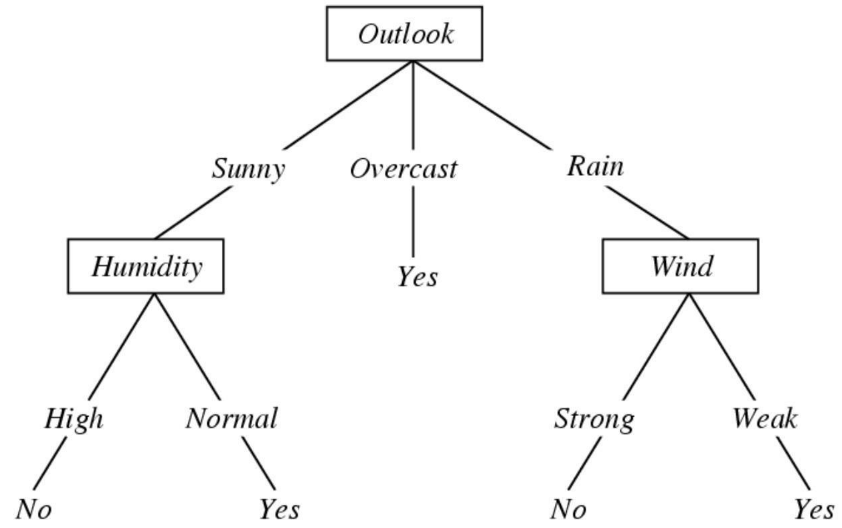


How to learn a decision tree from data?

Learning in DT

- What questions to ask
- In what order to ask them
- What answer to predict once you have asked enough questions.

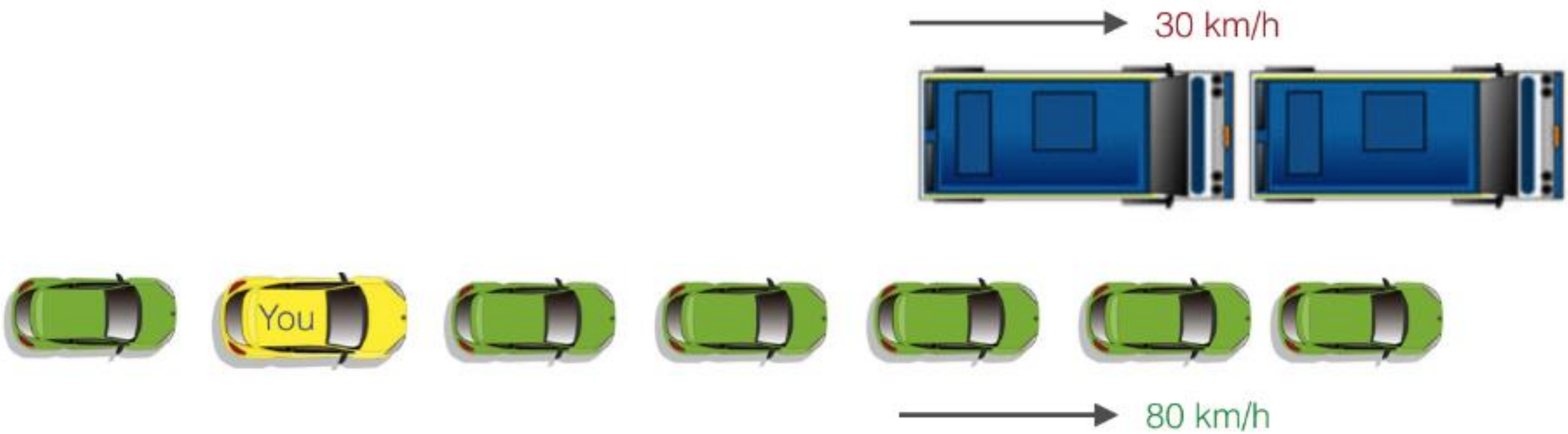
- Questions ➔ Features
- Answers ➔ Feature values
- Classification at leaf: Label



How?

- SO MANY possible DTs!
  - exhaustive search infeasible!
- We will use a heuristic ***greedy*** search algorithm.

Greedy Algorithm …



How?

- SO MANY possible DTs!
  - exhaustive search infeasible!
- We will use a heuristic ***greedy*** search algorithm.
  - Pick questions to ask, in order
  - Such that classification accuracy is maximized

If I could only ask one question, what question would I ask?

- You want to find a feature that is *most useful* in helping you guess whether to play tennis or not.

How to select the "best" feature?

- A good feature is a feature that lets us make correct classification decision
- Criteria to measure how good a feature is
  - classification accuracy
  - entropy
  - …

Approach ...

- Consider the entire training data.
- Look at the **histogram** of labels for each feature.
  - For **each value** of the feature, build a histogram over the labels.
- Now, suppose you were to ask the (feature) question on a random example and observe a value and you must immediately guess the label for this example.
  - Guess the majority label!
  - How well would I have done? In particular, how many examples would I classify correctly?
- Let's try it on the PlayTennis dataset

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	-

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

T(emperature): H(ot), M(ild), C(ool)

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

W(ind): S(trong), W(eak)

Overall …

	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	-

- **Overall**
  - **Acc(overall) = 9/14**

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	-

- **Overall**
  - **Acc(overall) = 9/14**
- **Outlook**

Outlook: Sunny …

	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	-

- **Overall**
  - **Acc(overall) = 9/14**
- **Outlook**
  - Acc(S)=3/5

Outlook: Overcast …

	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	-

- **Overall**
  - **Acc(overall) = 9/14**
- **Outlook**
  - Acc(S)=3/5
  - Acc(O)=4/4

Outlook: Rainy …

	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	-

- **Overall**
  - **Acc(overall) = 9/14**
- **Outlook**
  - Acc(S)=3/5
  - Acc(O)=4/4
  - Acc(R)=3/5

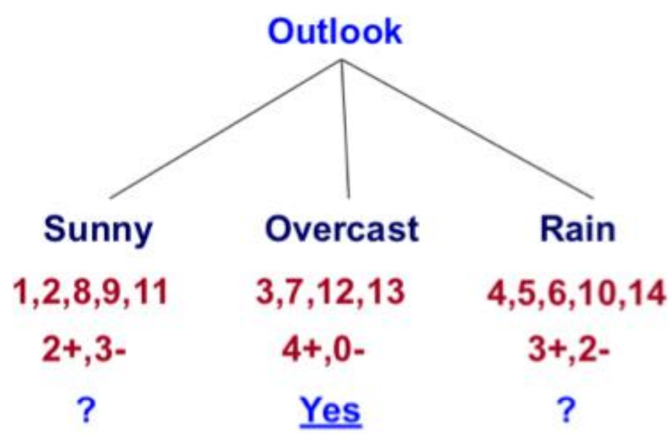
Outlook overall …

	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	-

- **Overall**
  - **Acc(overall) = 9/14**
- **Outlook**
  - Acc(S)=3/5
  - Acc(O)=4/4
  - Acc(R)=3/5
  - **Acc(Outlook)=10/14**
- **Other features?**

First split!

	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	-

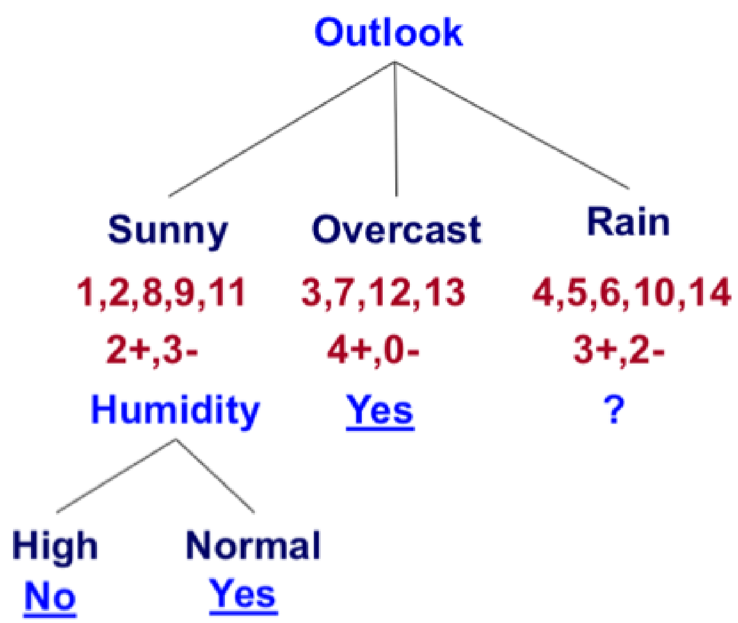


Continue until:

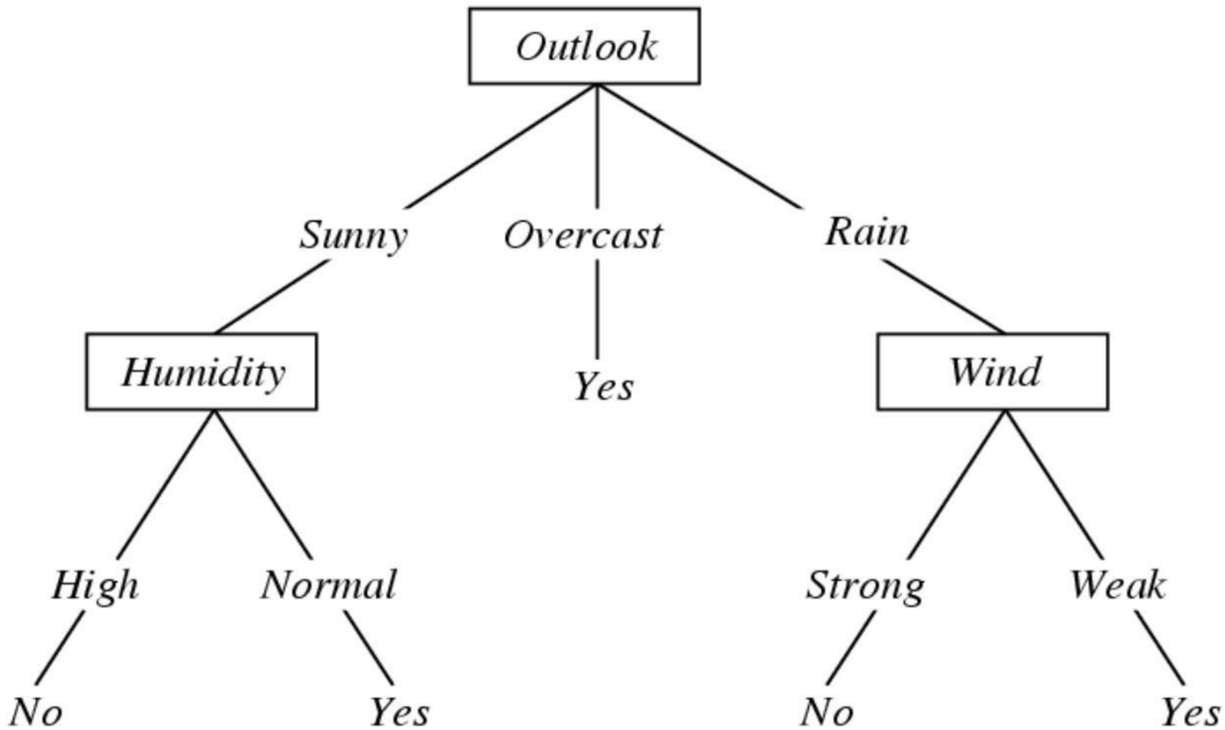
- All examples in the leaf have same label, or
- Every attribute is included in path

Sunny … Humidity?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
8	S	M	H	W	-
9	S	C	N	W	+
11	S	M	N	S	+



Full DT



Outlook
Sunny
Overcast
Rain
Humidity
Yes
Wind
High
Normal
Strong
Weak
No
Yes
No
Yes

DT Training Algorithm

Binary features

Algorithm 1 DecisionTreeTrain(*data*, *remaining features*)

```
guess ← most frequent answer in data                    // default answer for this data
if the labels in data are unambiguous then
    return Leaf(guess)                                  // base case: no need to split further
else if remaining features is empty then
    return Leaf(guess)                                  // base case: cannot split further
else                                                    // we need to query more features
    for all f ∈ remaining features do
        NO ← the subset of data on which f=no
        YES ← the subset of data on which f=yes
        score[f] ← # of majority vote answers in NO
                   + # of majority vote answers in YES
                              // the accuracy we would get if we only queried on f
    end for
    f ← the feature with maximal score(f)
    NO ← the subset of data on which f=no
    YES ← the subset of data on which f=yes
    left ← DecisionTreeTrain(NO, remaining features \ {f})
    right ← DecisionTreeTrain(YES, remaining features \ {f})
    return Node(f, left, right)
end if
```

Prediction with DT

Algorithm 2 DecisionTreeTest(*tree*, *test point*)

```
if tree is of the form Leaf(guess) then
    return guess
else if tree is of the form Node(f, left, right) then
    if f = no in test point then
        return DecisionTreeTest(left, test point)
    else
        return DecisionTreeTest(right, test point)
    end if
end if
```

Other feature selection criteria

- Entropy
- Information Gain
- Gini index