Decision Trees

Alice Gao

Lecture 7

Readings: RN 18.3, PM 7.3.1.

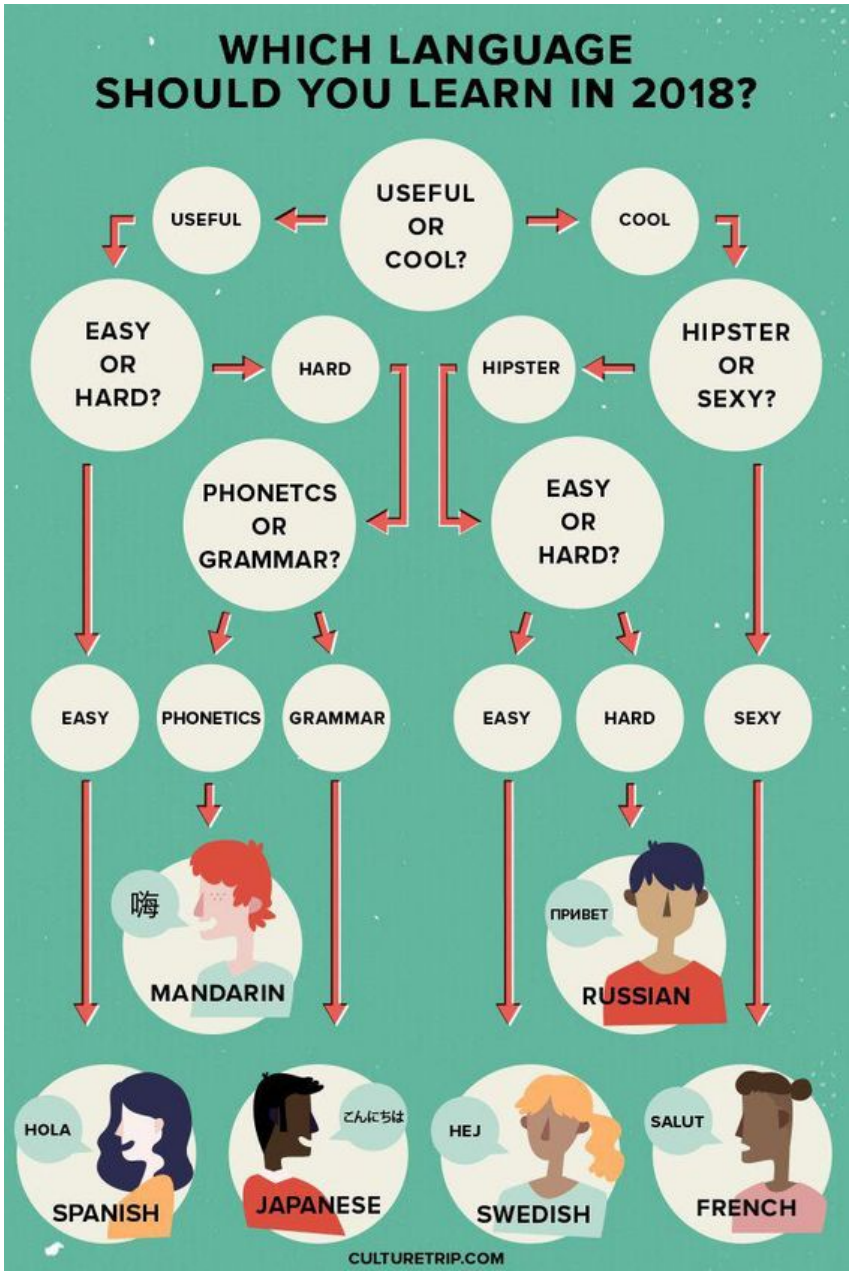
Outline

Learning Goals

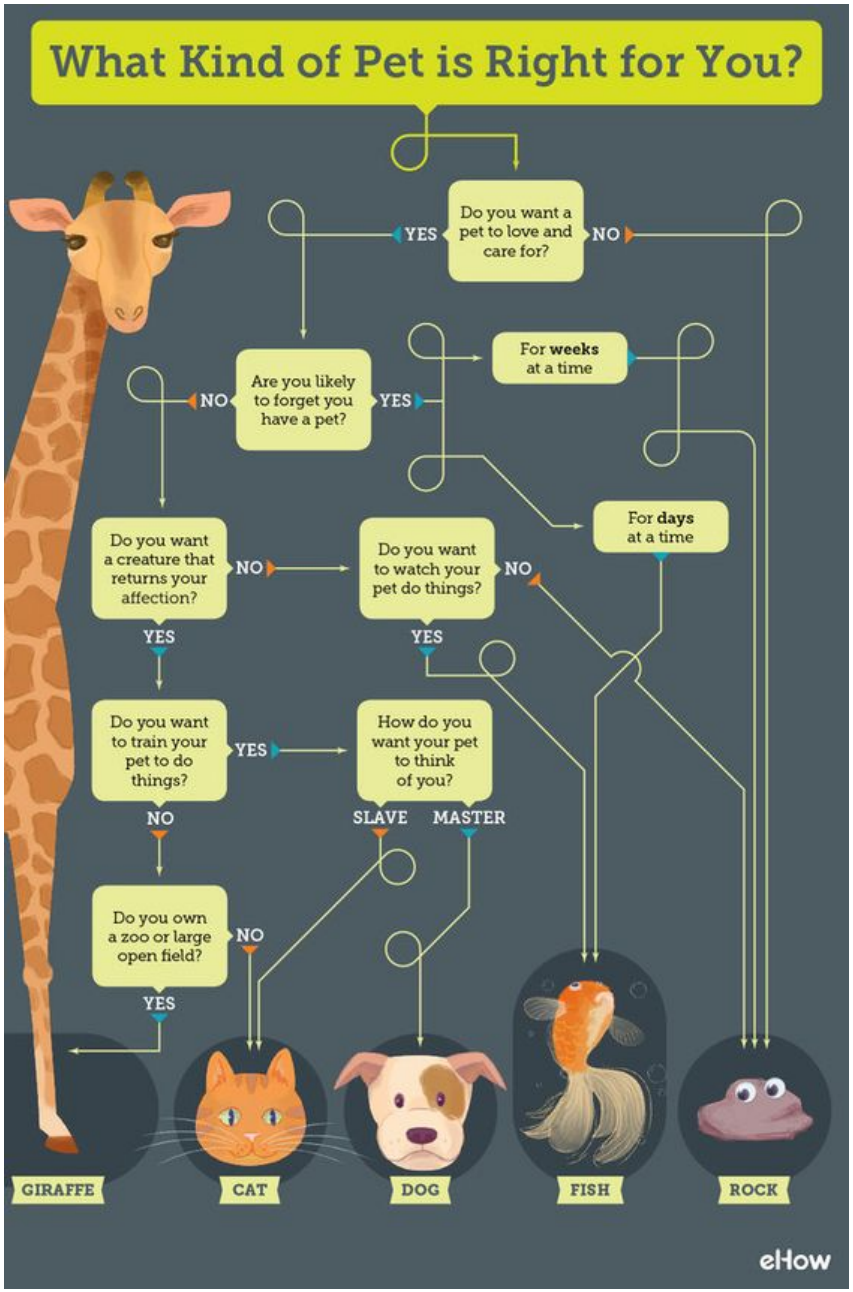
By the end of the lecture, you should be able to

- Describe the components of a decision tree.
- Construct a decision tree given an order of testing the features.
- Determine the prediction accuracy of a decision tree on a test set.
- Compute the entropy of a probability distribution.
- Compute the expected information gain for selecting a feature.
- Trace the execution of and implement the ID3 algorithm.

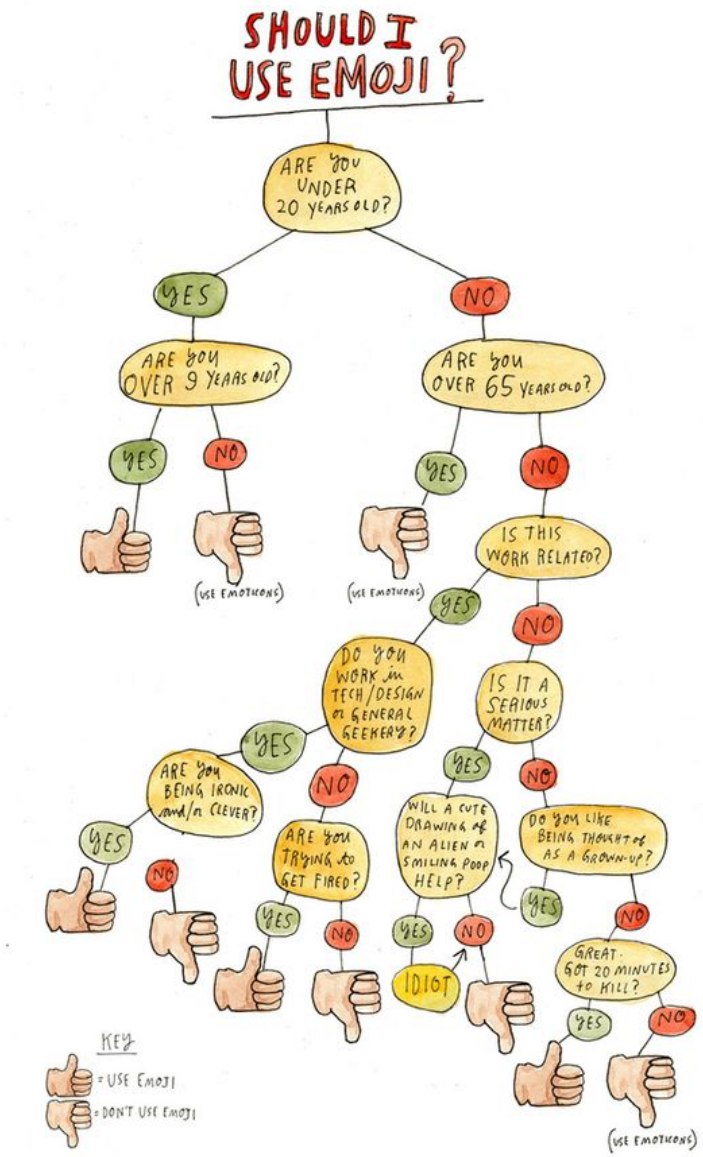
Which language should you learn?



What kind of pet is right for you?



Should I use emoji?



Jeeves the valet

Jeeves is a valet to Bertie Wooster. On some days, Bertie likes to play tennis and asks Jeeves to lay out his tennis things and book the court. Jeeves would like to predict whether Bertie will play tennis (and so be a better valet). Each morning over the last two weeks, Jeeves has recorded whether Bertie played tennis on that day and various attributes of the weather (training set).

Jeeves would like to evaluate the classifier he has come up with for predicting whether Bertie will play tennis. Each morning over the next two weeks, Jeeves records the following data (test set).

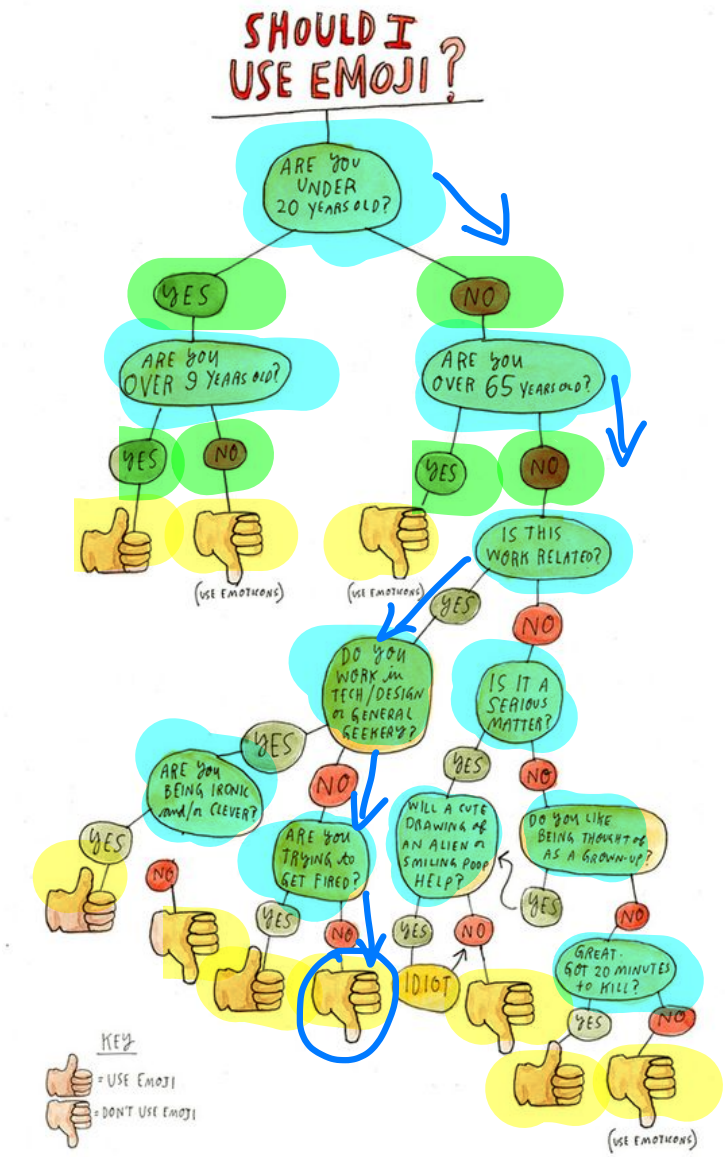
Jeeves the valet - training set

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Jeeves the valet - test set

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Mild	High	Strong	No
2	Rain	Hot	Normal	Strong	No
3	Rain	Cool	High	Strong	No
4	Overcast	Hot	High	Strong	Yes
5	Overcast	Cool	Normal	Weak	Yes
6	Rain	Hot	High	Weak	Yes
7	Overcast	Mild	Normal	Weak	Yes
8	Overcast	Cool	High	Weak	Yes
9	Rain	Cool	High	Weak	Yes
10	Rain	Mild	Normal	Strong	No
11	Overcast	Mild	High	Weak	Yes
12	Sunny	Mild	Normal	Weak	Yes
13	Sunny	Cool	High	Strong	No
14	Sunny	Cool	High	Weak	No

What is a decision tree?



What is a decision tree?

- A simple model for supervised classification.
- A single discrete target feature.
- Each internal node performs a Boolean test on an input feature.
- The edges are labeled with values of the input feature.
- Each leaf node specifies a value for the target feature.

Classify an example using a decision tree

- Go down the tree, evaluate each test and follow the corresponding edge.
- When a leaf is reached, return the classification on that leaf.

Should I use emoji?

- I am 30 years old.
- This is work related.
- I am an accountant.
- I am not trying to get fired.

If we convert a decision tree to a program, what does it look like?

Issues in learning a decision tree

How should we build a decision tree?

- We need to determine an order of testing the input features.
- Given an order of testing the input features, we can build a decision tree by splitting the examples.

Which decision tree should we generate?

- Which order of testing the input features should we use?
  - The search space is too big for systematic search.
  - Solution: greedy (myopic) search.
- Should we grow a full tree or not?
  - A decision tree can represent any discrete function of input features.
  - Need a bias. For example, prefer the smallest tree. (Least depth? Fewest nodes?)

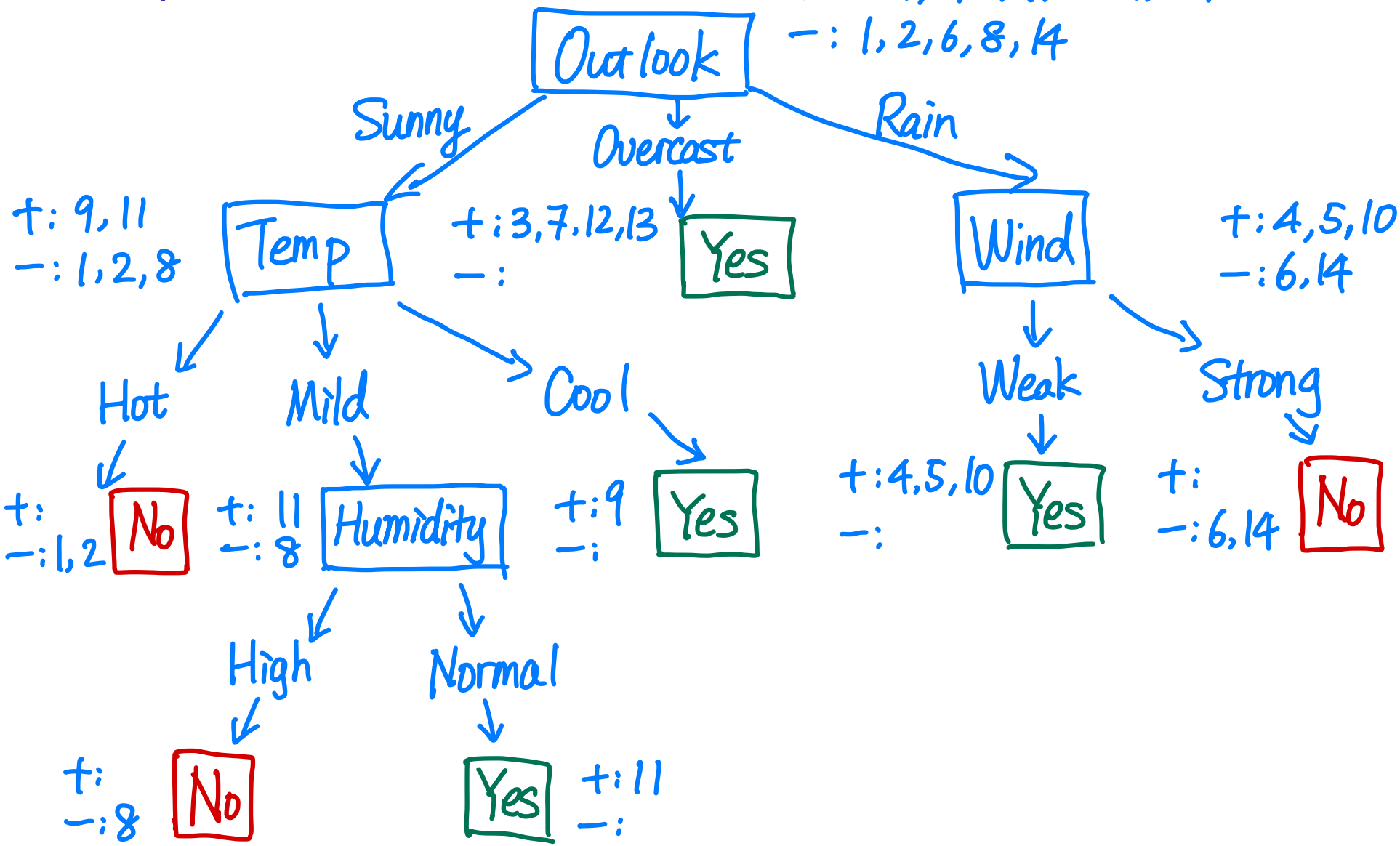
Construct a Decision Tree Given an Order of Testing Features

Construct a decision tree
using the following order of testing features.

- First, test Outlook.
- For Outlook = Sunny, test Temp.
- For Outlook = Rain, test Wind.
- For other branches, test Humidity before testing Wind.

Let's draw the decision tree.

Example: Decision tree



When do we stop?

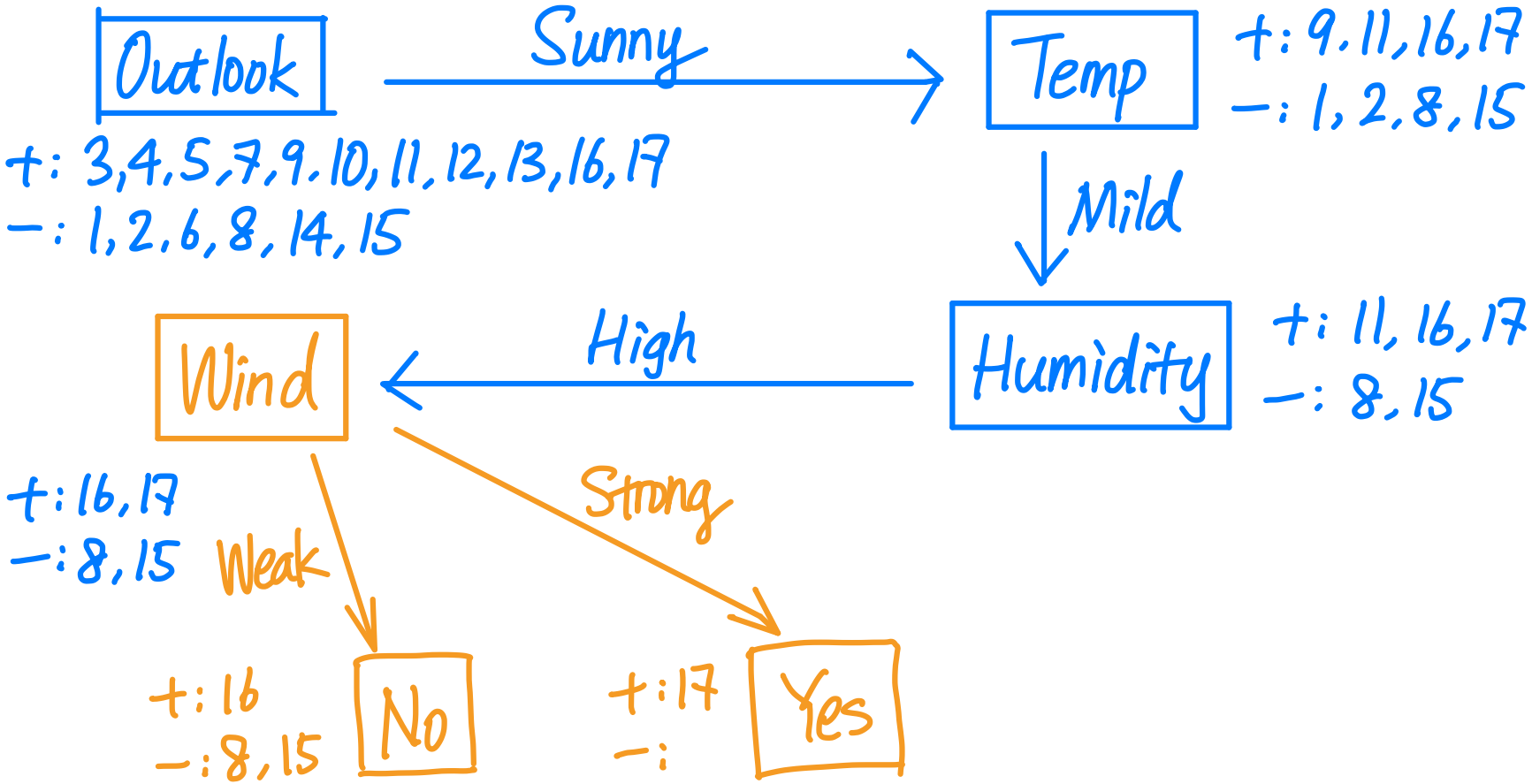
- All the examples belong to the same class.
- There are no more features to test.
- There are no more examples.

No features left

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No
15	Sunny	Mild	High	Weak	No
16	Sunny	Mild	High	Weak	Yes
17	Sunny	Mild	High	Strong	Yes

No features left

Complete the branch of the tree where
Outlook = Sunny, Temp = Mild, and Humidity = High.



No features left

Why do we have this special case?

The data is noisy. Even if we know the values of all the input features, we still cannot make a deterministic decision. The target feature may be influenced by an unobserved input feature.

What are some possible ways of handling this case?

- Predict the majority class, or
- Make a probabilistic decision.

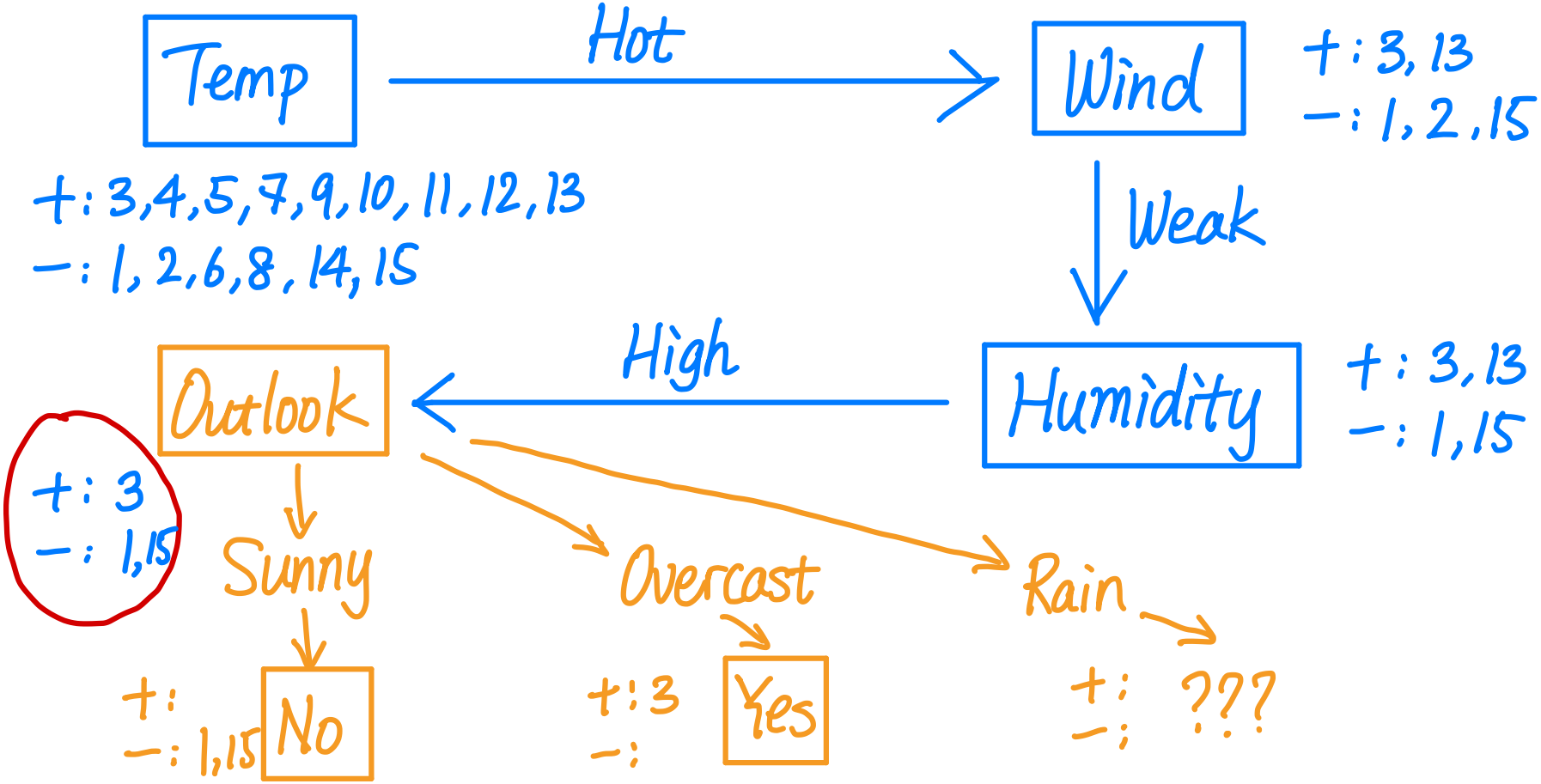
No examples left

Consider the revised training set below.

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No
15	Sunny	Hot	High	Weak	No

No examples left

Complete the branch of the tree where
Temp = Hot, Wind = Weak, and Humidity = High.



No examples left

Why do we have this special case?

A combination of features is not present in the training set. If we never observe this combination, we don't know how to predict it.

What are some possible ways of handling this case?

Use the examples at the parent node.

Make a majority decision or a probabilistic decision.

The Decision Tree Learner algorithm

Algorithm 1 Decision Tree Learner (examples, features)

```
1:  if all examples are in the same class then
2:      return the class label.
3:  else if no features left then
4:      return the majority decision.
5:  else if no examples left then
6:      return the majority decision at the parent node.
7:  else
8:      choose a feature f.
9:      for each value v of feature f do
10:         build edge with label v.
11:         build sub-tree using examples where the value of f is v.
```

base cases

recursive case

data is noisy.

input combo is absent in the training set.

split up examples based on values of the feature.

build sub-trees recursively.

Which feature should we test at each step?

Which feature should we test at each step?

- How can we create a shallow/small tree?
  Need to minimize the number of tests.

- Finding the optimal order of testing features is difficult.
  Greedy search — make the best myopic choice at each step.

- At each step, test a feature that
  makes the biggest difference to the classification.

Identifying the most important feature

- We want to make a decision as soon as possible
  — reduce uncertainty as much as possible
- Information content of a feature
  = uncertainty before testing the feature
  - uncertainty after testing the feature
- Measure uncertainty using the notion of entropy.

Given a distribution $P(c_1), ..., P(c_k)$ over k outcomes $c_1, ..., c_k$, the entropy of the distribution is

$$I(P(c_1), ..., P(c_k)) = -\sum_{i=1}^{k} P(c_i) \log_2(P(c_i))$$

CQ: Entropy of a distribution over two outcomes

CQ: What is the entropy of the distribution $(0.5, 0.5)$?

$$I(P(c_1), ..., P(c_k)) = -\sum_{i=1}^{k} P(c_i) \log_2(P(c_i))$$

(A) 0.2
(B) 0.4
(C) 0.6
(D) 0.8
(E) 1

$-(0.5 * \log_2(0.5)) * 2 = 1$

a lot of uncertainty.

CQ: Entropy of a distribution over two outcomes

CQ: What is the entropy of the distribution $(0.01, 0.99)$?

$$I(P(c_1), ..., P(c_k)) = -\sum_{i=1}^{k} P(c_i) \log_2(P(c_i))$$

(A) 0.02
(B) 0.04
(C) 0.06
(D) 0.08
(E) 0.1

$-(0.01 * \log_2(0.01) + 0.99 * \log_2(0.99))$

$= -(-0.0664 - 0.01435)$

≈ 0.08

very little uncertainty.

We almost know for sure that the second outcome will be realized.

Entropy of a distribution over two outcomes

Consider a distribution $(p, 1-p)$ where $0 \leq p \leq 1$.

- What is the maximum entropy of this distribution?

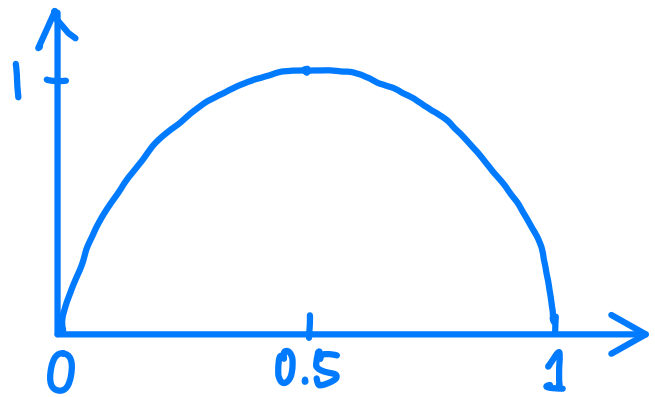
  1 achieved at $p = 0.5$.

- What is the minimum entropy of this distribution?

  0 achieved at $p = 0$ and $p = 1$.

  Define
  $I(0,1) = 0$
  $I(1,0) = 0$

- Plot the entropy of the distribution $(p, 1-p)$ with respect to p.



$0 * \log_2 0 + 1 * \log_2 1$

undefined.

Expected information gain of testing a feature

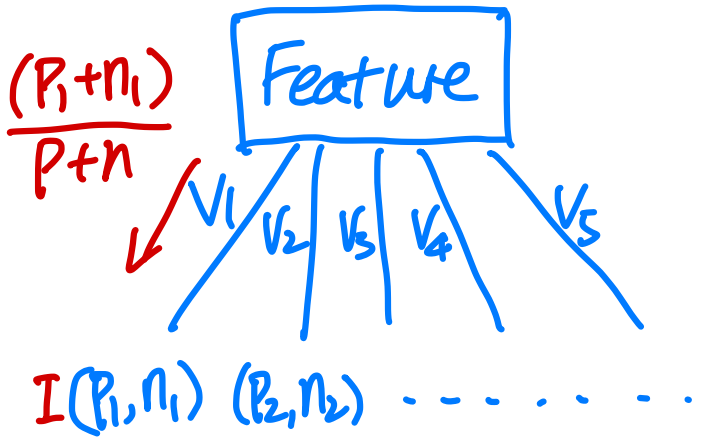
- The feature has k values $v_1, ..., v_k$.
- Before testing the feature,
  we have p positive and n negative examples.
- After testing the feature,
  for each value v_i of the feature,
  we have p_i positive and n_i negative examples.

Let's calculate the expected information gain of testing this feature.

Expected information gain of testing a feature

Entropy before testing the feature:

$$H_{before} = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right)$$



Expected entropy after testing the feature:

$$H_{after} = \sum_{i=1}^{k} \frac{p_i+n_i}{p+n} * I\left(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}\right)$$

Expected information gain (entropy reduction) is

$$InfoGain = H_{before} - H_{after}$$

Expected information gain of testing a feature

Entropy before testing the feature:

$$H_{before} = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right)$$

Expected entropy after testing the feature:

$$H_{after} = \sum_{i=1}^{k} \frac{p_i + n_i}{p+n} * I\left(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}\right)$$

Expected information gain (entropy reduction) is

$$\text{Info Gain} = H_{before} - H_{after}$$

can be generalized to more than 2 classes.

CQ: Entropy and information gain

$\log_2 x = \frac{\ln x}{\ln 2}$

CQ: What is the entropy of the examples before we select a feature for the root node of the tree?

(A) 0.54
(B) 0.64
(C) 0.74
(D) 0.84
(E) 0.94

14 examples. 9 +ve. 5 −ve.

$$-\left(\frac{9}{14}\log_2\left(\frac{9}{14}\right) + \frac{5}{14}\log_2\left(\frac{5}{14}\right)\right)$$

$$= -\left(\frac{9}{14}\cdot(-0.637) + \frac{5}{14}\cdot(-1.485)\right)$$

$$= -(-0.939)$$

$$= 0.939$$

CQ: Entropy and information gain

CQ: What is the expected information gain if we select **Outlook** as the root node of the tree?

(A) 0.237

(B) 0.247

(C) 0.257

(D) 0.267

(E) 0.277

Outlook = Sunny 2+ve 3−ve. 5

= Overcast 4+ve 0−ve. 4

= Rain 3+ve 2−ve. 5

$$Gain(Outlook) = 0.94 - \left(\frac{5}{14} I\left(\frac{2}{5}, \frac{3}{5}\right) + \frac{4}{14} I\left(\frac{4}{4}, \frac{0}{4}\right) + \frac{5}{14} I\left(\frac{3}{5}, \frac{2}{5}\right)\right)$$

$$= 0.94 - \left(\frac{5}{14} 0.971 + \frac{4}{14} 0 + \frac{5}{14} 0.971\right) = 0.94 - 0.694 = 0.247$$

CQ: Entropy and information gain

CQ: What is the expected information gain if we select **Humidity** as the root node of the tree?

(A) 0.151
(B) 0.251
(C) 0.351
(D) 0.451
(E) 0.551

Humidity = Normal 6 +ve 1 −ve

= High 3 +ve 4 −ve

Gain(Humidity)

$= 0.940 - \left(\frac{7}{14} I\left(\frac{3}{7}, \frac{4}{7}\right) + \frac{7}{14} I\left(\frac{6}{7}, \frac{1}{7}\right)\right)$

$= 0.940 - \left(\frac{7}{14} 0.985 + \frac{7}{14} 0.592\right)$

$= 0.940 - 0.789$

$= 0.151$

Root node: Gain(Outlook) = 0.247 Gain(Humidity) = 0.151

Gain(Temp) = 0.029 Gain(Wind) = 0.048

Outlook = Sunny +: 9, 11 −: 1, 2, 8

Temp = Hot +: −: 1, 2 Mild +: 11 −: 8 Cool +: 9 −:

$$Gain = I\left(\frac{2}{5},\frac{3}{5}\right) - \left(\frac{2}{5}I\left(\frac{0}{2},\frac{2}{2}\right) + \frac{2}{5}I\left(\frac{1}{2},\frac{1}{2}\right) + \frac{1}{5}I\left(\frac{1}{1},\frac{0}{1}\right)\right)$$

$$= 0.97 - \left(\frac{2}{5}\cdot 0 + \frac{2}{5}\cdot 1 + \frac{1}{5}\cdot 0\right) = 0.97 - 0.4 = 0.57$$

Humidity = High: +: −: 1, 2, 8 Normal: +: 9, 11 −:

$$Gain = I\left(\frac{2}{5},\frac{3}{5}\right) - \left(\frac{3}{5}I\left(\frac{0}{3},\frac{3}{3}\right) + \frac{2}{5}I\left(\frac{2}{2},\frac{0}{2}\right)\right)$$

$$= 0.97 - \left(\frac{3}{5}\cdot 0 + \frac{2}{5}\cdot 0\right) = 0.97 - 0 = 0.97$$

Wind = Weak (+: 9, −: 1, 8) Strong (+: 11, −: 2,)

$$Gain = I\left(\frac{2}{5},\frac{3}{5}\right) - \left(\frac{3}{5}I\left(\frac{1}{3},\frac{2}{3}\right) + \frac{2}{5}I\left(\frac{1}{2},\frac{1}{2}\right)\right)$$

$$= 0.97 - \left(\frac{3}{5}\cdot 0.918 + \frac{2}{5}\cdot 1\right) = 0.97 - 0.951 = 0.019$$

Outlook = Rain (+: 4, 5, 10; −: 6, 14)

Temp = Hot (+: none; −: none) Mild (+: 4, 10; −: 14) Cool (+: 5,; −: 6,)

$$Gain = I\left(\frac{3}{5},\frac{2}{5}\right) - \left(\frac{3}{5}I\left(\frac{2}{3},\frac{1}{3}\right) + \frac{2}{5}I\left(\frac{1}{2},\frac{1}{2}\right)\right)$$

$$= 0.97 - \left(\frac{3}{5}0.918 + \frac{2}{5}\cdot 1\right) = 0.97 - 0.951 = 0.019$$

Humidity = Normal $\begin{matrix} +: 5, 10 \\ -: 6 \end{matrix}$ High: $\begin{matrix} +: 4. \\ -: 14. \end{matrix}$

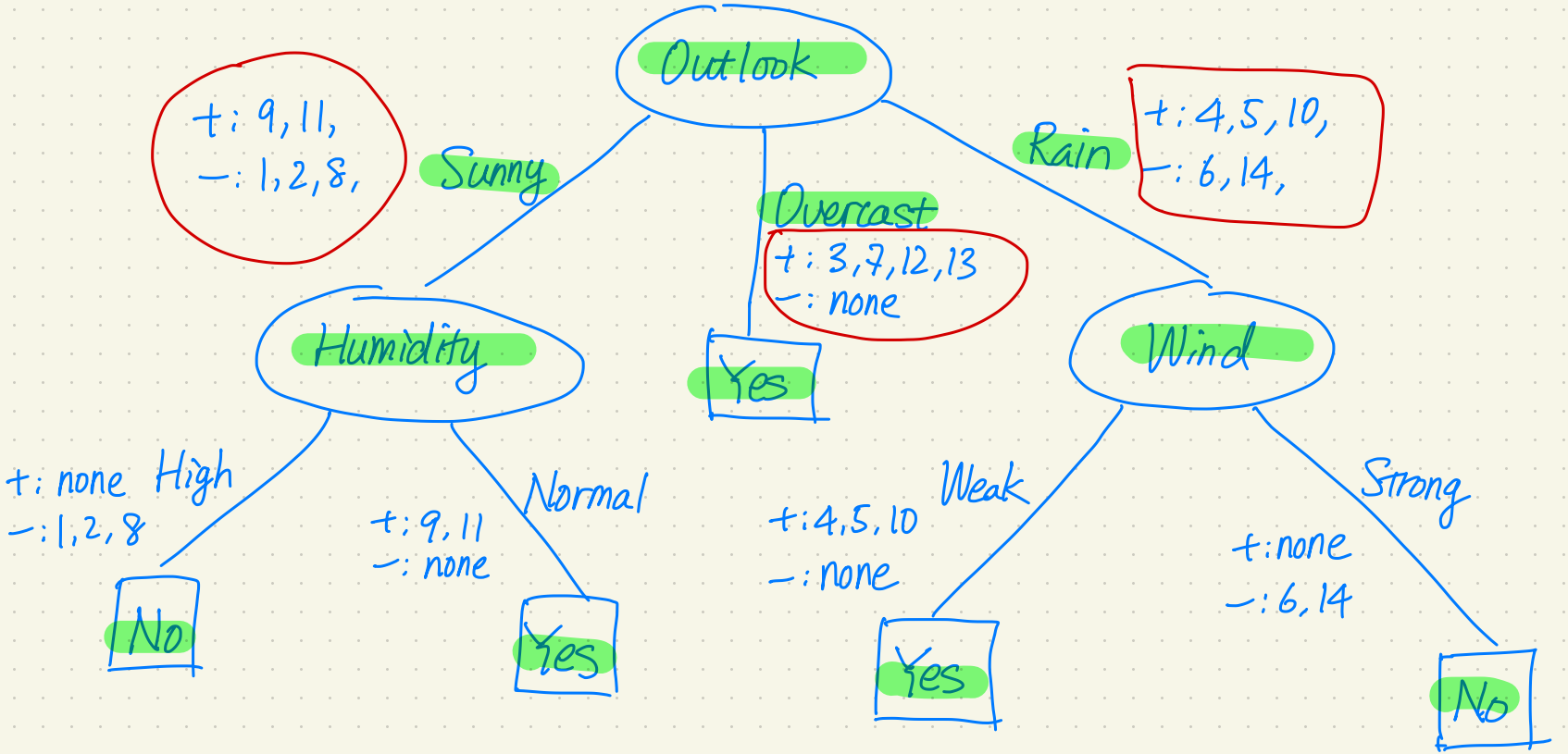
$$Gain = I\left(\frac{3}{5}, \frac{2}{5}\right) - \left(\frac{3}{5} I\left(\frac{2}{3}, \frac{1}{3}\right) + \frac{2}{5} I\left(\frac{1}{2}, \frac{1}{2}\right)\right)$$

$$= 0.971 - \left(\frac{3}{5} \cdot 0.918 + \frac{2}{5} \cdot 1\right) = 0.971 - 0.951 = 0.019$$

Wind = Weak $\begin{matrix} +: 4, 5, 10 \\ -: \text{none} \end{matrix}$ Strong: $\begin{matrix} +: \text{none} \\ -: 6, 14 \end{matrix}$

$$Gain = I\left(\frac{3}{5}, \frac{2}{5}\right) - \left(\frac{2}{5} I\left(\frac{0}{2}, \frac{2}{2}\right) + \frac{3}{5} I\left(\frac{3}{3}, \frac{0}{3}\right)\right)$$

$$= 0.971 - \left(\frac{2}{5} \cdot 0 + \frac{3}{5} \cdot 0\right) = 0.971 - 0 = 0.971$$


Outlook
+: 9,11,
−: 1,2,8,
Sunny
Overcast
+: 3,7,12,13
−: none
Rain
+: 4,5,10,
−: 6,14,
Humidity
Yes
Wind
+: none
−: 1,2,8
High
+: 9,11
−: none
Normal
+: 4,5,10
−: none
Weak
+: none
−: 6,14
Strong
No
Yes
Yes
No

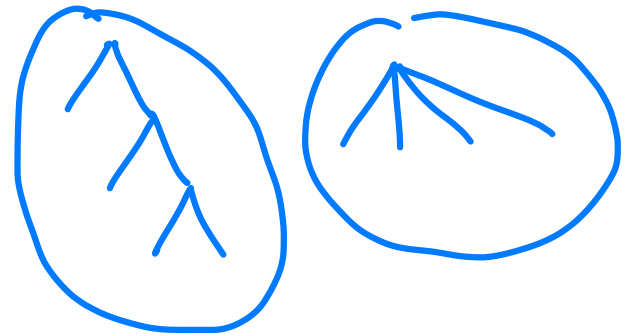
Jeeves dataset with real-valued temperatures

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	29.4	High	Weak	No
2	Sunny	26.6	High	Strong	No
3	Overcast	28.3	High	Weak	Yes
4	Rain	21.1	High	Weak	Yes
5	Rain	20.0	Normal	Weak	Yes
6	Rain	18.3	Normal	Strong	No
7	Overcast	17.7	Normal	Strong	Yes
8	Sunny	22.2	High	Weak	No
9	Sunny	20.6	Normal	Weak	Yes
10	Rain	23.9	Normal	Weak	Yes
11	Sunny	23.9	Normal	Strong	Yes
12	Overcast	22.2	High	Strong	Yes
13	Overcast	27.2	Normal	Weak	Yes
14	Rain	21.7	High	Strong	No

Jeeves dataset ordered by temperatures

Day	Outlook	Temp	Humidity	Wind	Tennis?
7	Overcast	17.7	Normal	Strong	Yes
6	Rain	18.3	Normal	Strong	No
5	Rain	20.0	Normal	Weak	Yes
9	Sunny	20.6	Normal	Weak	Yes
4	Rain	21.1	High	Weak	Yes
14	Rain	21.7	High	Strong	No
8	Sunny	22.2	High	Weak	No
12	Overcast	22.2	High	Strong	Yes
10	Rain	23.9	Normal	Weak	Yes
11	Sunny	23.9	Normal	Strong	Yes
2	Sunny	26.6	High	Strong	No
13	Overcast	27.2	Normal	Weak	Yes
3	Overcast	28.3	High	Weak	Yes
1	Sunny	29.4	High	Weak	No

Handling a discrete feature



Trade off:
- complexity of the tree
- depth of the tree

- Binary splits only.

  tree is simpler and more compact.
  tree tends to be deeper.

- Allow multi-way splits.
  - tree becomes more complex than if-then-else.
  - tree tends to be shallower.
  - info gain prefers a var w/ a larger domain.

Handling a real-valued feature

- Discretize the feature.

  pros: easy to do

  cons: lose valuable info.
  may make tree complex.

- Always do binary tests. Dynamically choose a split point.

  multi-way splits problematic w/ unbounded domain.
  w/ binary splits, may test the feature many time.
  the tree may become much deeper.

Choosing a split point for a real-valued feature

1. Sort the instances according to the real-valued feature
2. Possible split points are values
   that are midway between two different values.
3. Suppose that the feature changes from X to Y. Should we consider $(X + Y)/2$ as a possible split point?
4. Let L_X be all the labels for the examples
   where the feature takes the value X.
5. Let L_Y be all the labels for the examples
   where the feature takes the value Y.
6. If there exists a label $a \in L_X$ and a label $b \in L_Y$ such that $a \neq b$, then $(X + Y)/2$ is a possible split point.
7. Determine the expected information gain for each possible split point and choose the split point with the largest gain.

CQ: Is this a possible split point? (1/3)

CQ: For the Jeeves training set,
is the midpoint between 20.0 and 20.6 a possible split point?

(A) Yes.

(B) No.

CQ: Is this a possible split point? (2/3)

CQ: For the Jeeves training set,
is the midpoint between 21.1 and 21.7 a possible split point?

(A) Yes.

(B) No.

CQ: Is this a possible split point? (3/3)

CQ: For the Jeeves training set,
is the midpoint between 21.7 and 22.2 a possible split point?

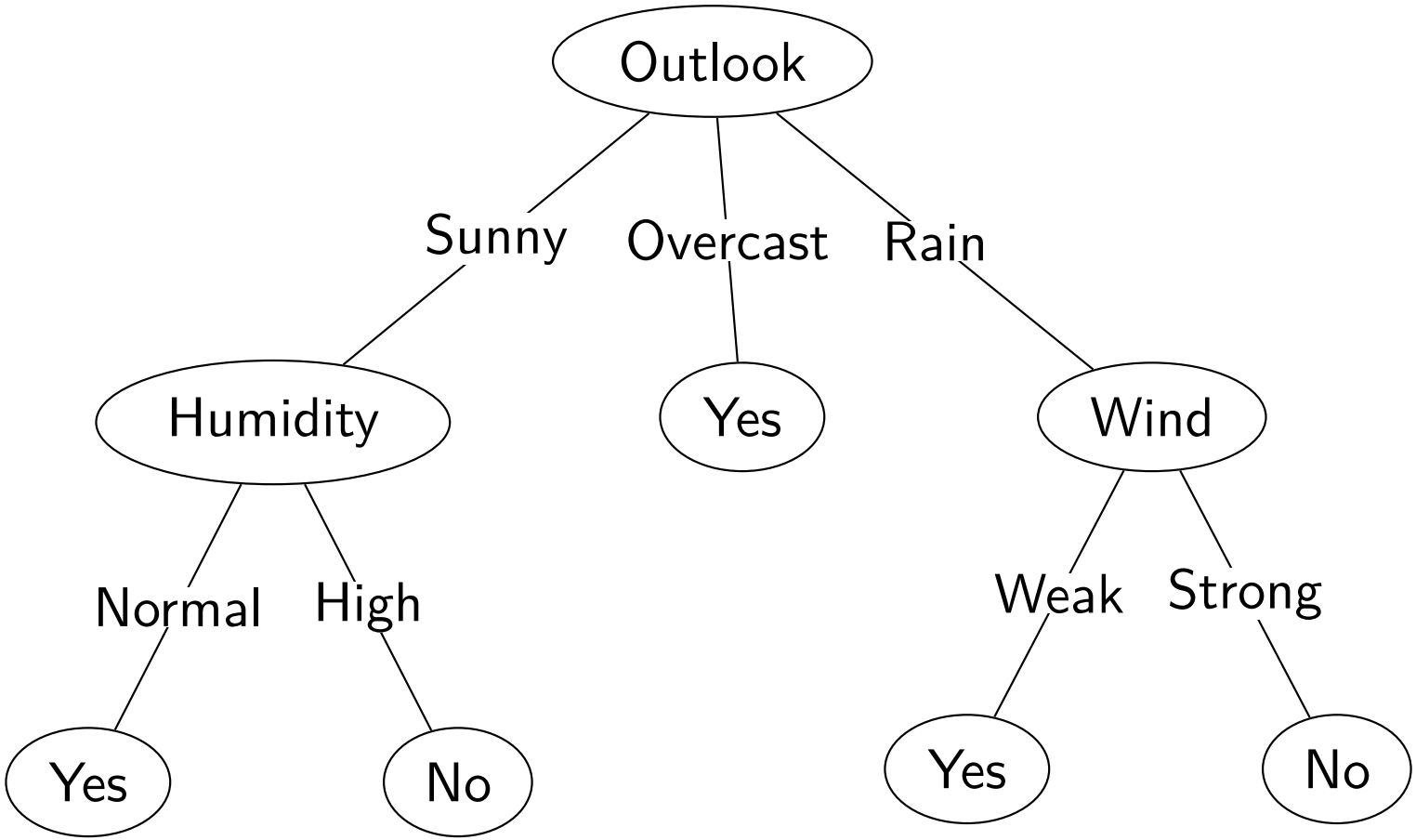
(A) Yes.

(B) No.

Jeeves the valet - training set

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Decision tree generated by the learner algorithm

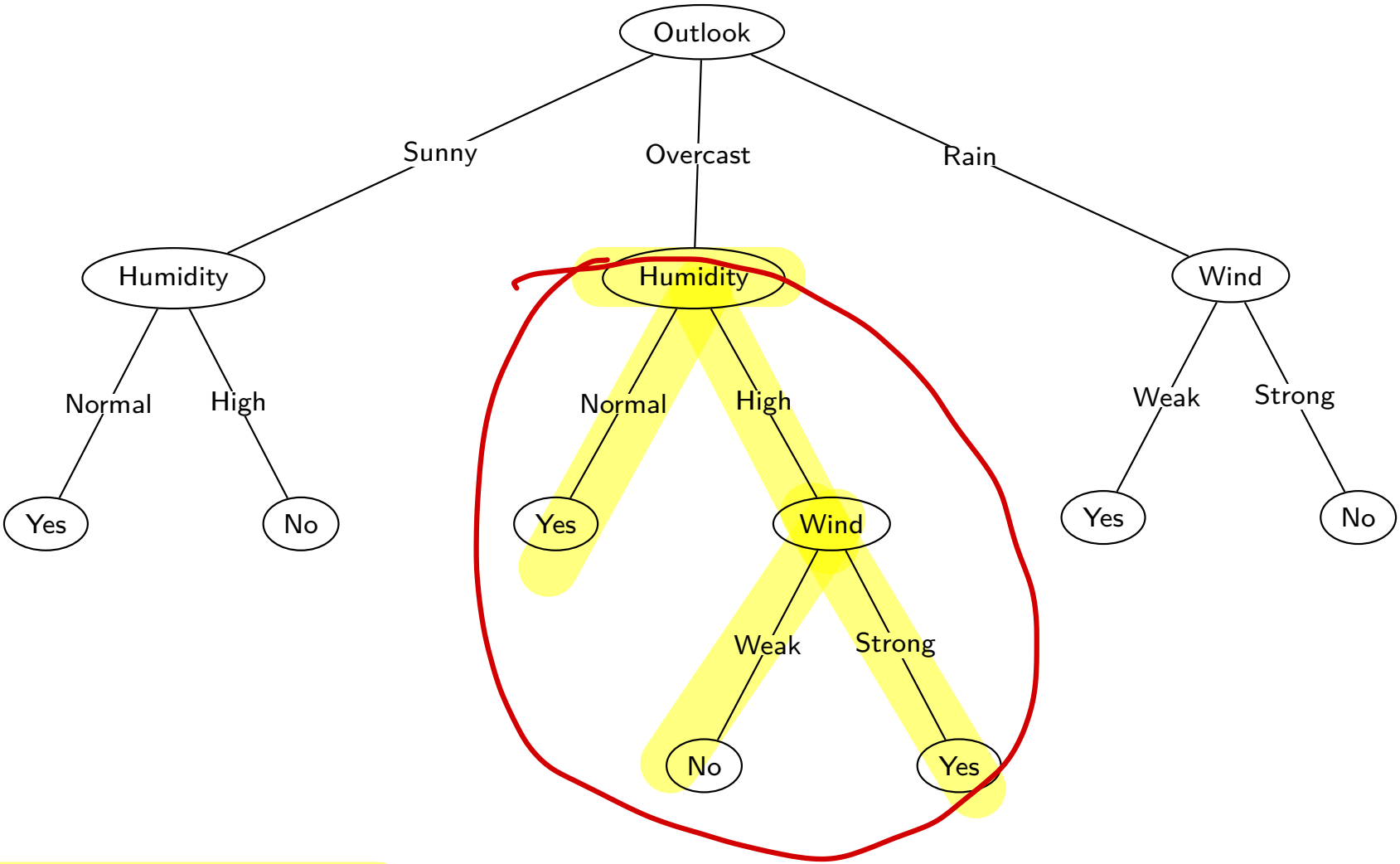


Test error is 0/14.

Jeeves the valet - a corrupted training set

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	**No**
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Decision tree for the corrupted data set



Test error is 2/14.

Dealing with Over-fitting

Problem: Growing a full tree is likely to lead to over-fitting.

Strategies to prevent over-fitting:

- Minimum examples at a node.
- Information gain must be above a threshold.
- Maximum depth.
- Post-prune the tree.

Revisiting the Learning Goals

By the end of the lecture, you should be able to

- Describe the components of a decision tree.
- Construct a decision tree given an order of testing the features.
- Determine the prediction accuracy of a decision tree on a test set.
- Compute the entropy of a probability distribution.
- Compute the expected information gain for selecting a feature.
- Trace the execution of and implement the algorithm for learning a decision tree.
- Construct decision trees with real-valued features.
- Construct a decision tree to prevent over-fitting.