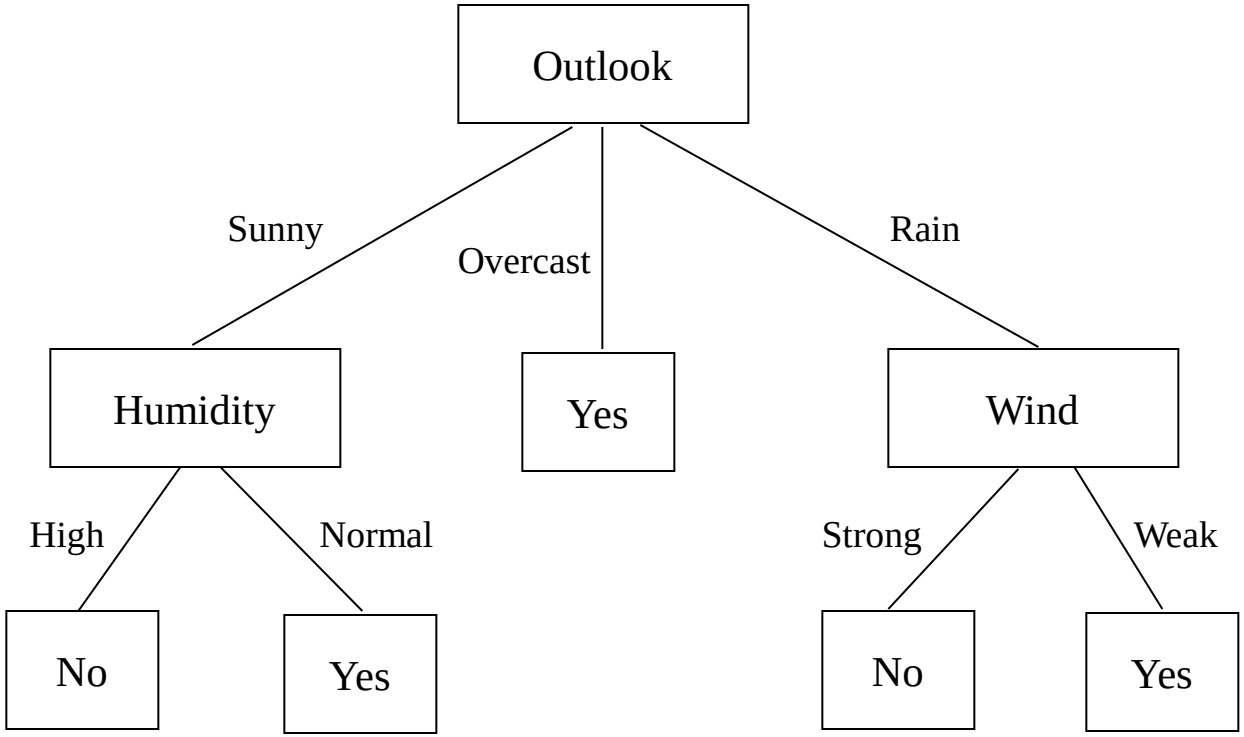
Decision trees

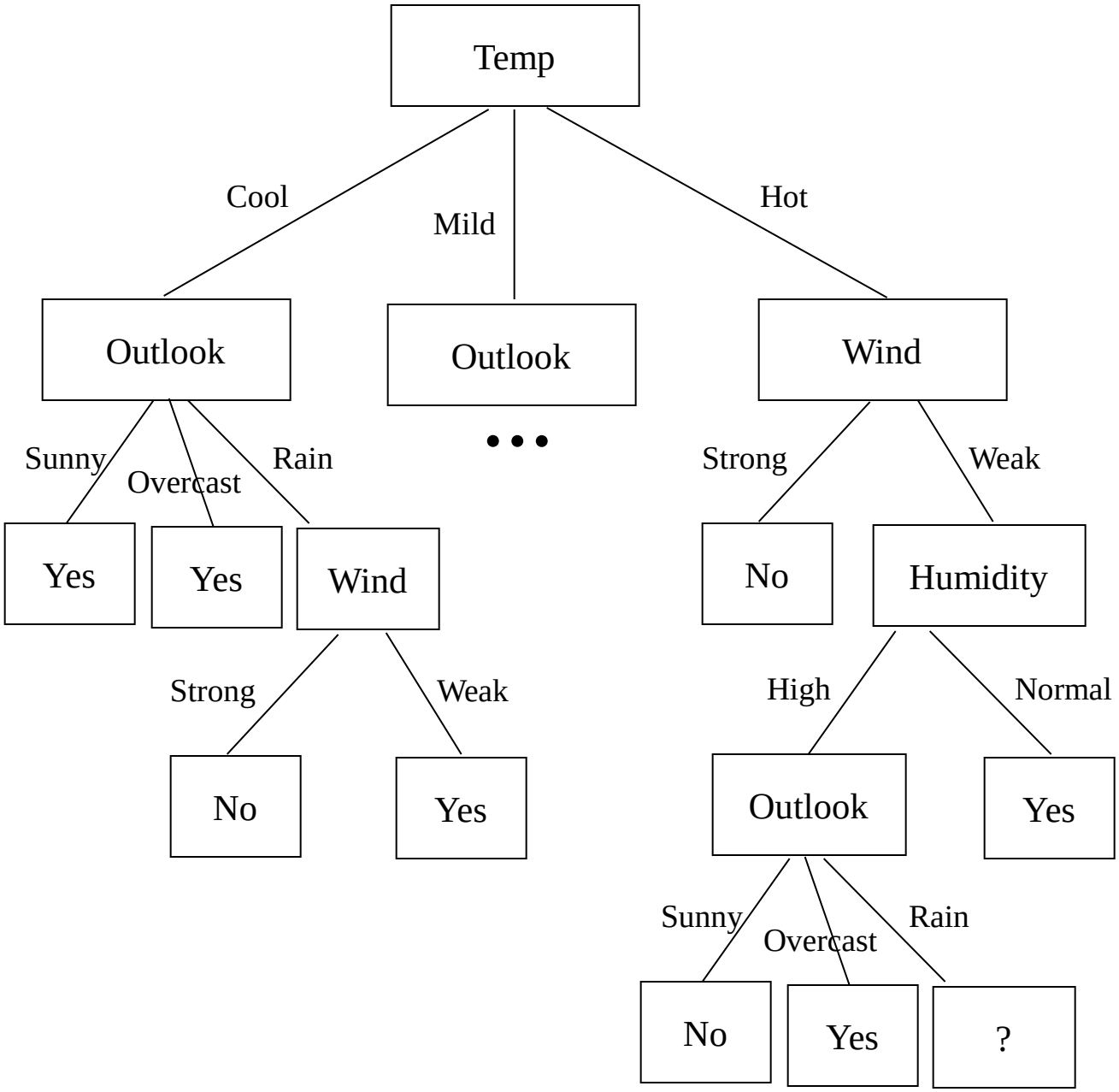
- Start at the root
- At each node in the tree, a feature is tested
  - arcs are labeled with the values of the feature
- Leaves contain the classification

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 be able 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.

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

Can Jeeves learn to predict Bertie's tennis playing?


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


Temp
Cool
Mild
Hot
Outlook
Outlook
Wind
Sunny
Overcast
Rain
Strong
Weak
Yes
Yes
Wind
No
Humidity
Strong
Weak
High
Normal
No
Yes
Outlook
Yes
Sunny
Overcast
Rain
No
Yes
?

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.

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

How well does Jeeves predict Bertie's tennis playing?

ID3 algorithm

Input: Set S of positive and negative examples

Set F of features

ID3(F, S)

1. if S contains only positive examples, return “yes”
2. if S contains only negative examples, return “no”
3. else

choose best feature $f \in F$

for each value v of f do

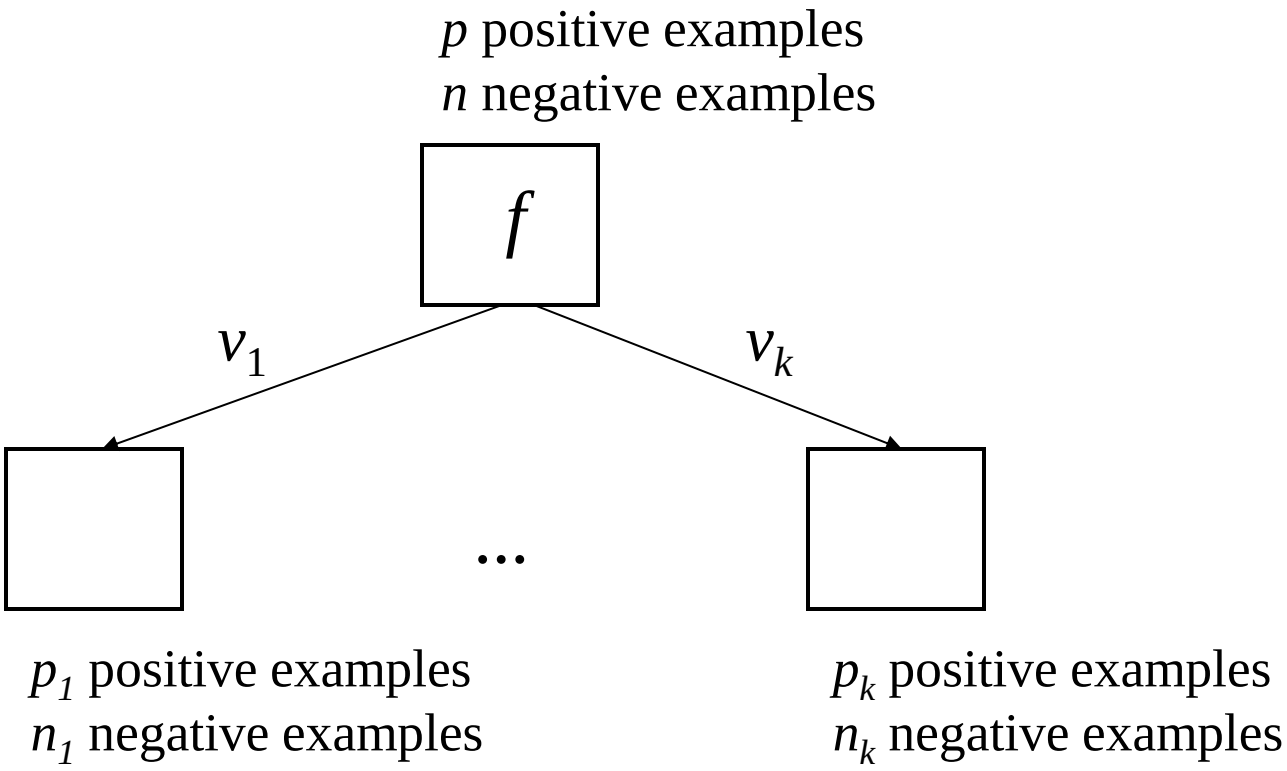
add arc to tree with label v

add subtree ID3(F - {f }, { $s \in S \mid f(s) = v$ })

Which is the *best* feature?

- Assumption: simplest decision tree generalizes best
  - NP-Complete in general to find smallest decision tree
- Approximate using greedy, heuristic approach
- Heuristics attempt to distinguish necessary features from extraneous features
  - simplest decision tree is least likely to include unnecessary feature tests

Choosing a feature



Worst feature: $p_1, \ldots, p_k, n_1, \ldots, n_k$, are all the same size

Best feature: $p_i = 0$ or $n_i = 0$, for all i

(divides examples into sets that are all positive or all negative)

Information theory

- Suppose there are l outcomes $c_1, \ldots, c_l$ each with probability $\mathrm{P}(c_1), \ldots, \mathrm{P}(c_l)$, respectively
- The information content of knowing the actual outcome is given by,

$$\mathrm{I}(\mathrm{P}(c_1), \ldots, \mathrm{P}(c_l)) = \sum_{i=1}^{l} -\mathrm{P}(c_i) \log_2(\mathrm{P}(c_i))$$

Information theory

- Consider case with two possible outcomes
  - positive, “yes”, “heads”
  - negative, “no”, “tails”
- By definition:

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

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

- More generally, consider $I(p, 1–p)$, for p in [0, 1]
  - maximum at $p = ½$

Examples of information gain

- Consider the flip of a fair coin

  $$I(\tfrac{1}{2}, \tfrac{1}{2}) = -\tfrac{1}{2}\log_2(\tfrac{1}{2}) - \tfrac{1}{2}\log_2(\tfrac{1}{2}) = 1 \text{ bit}$$

  Being told the result of the flip gives you 1 bit of information

- Consider the flip of an unfair coin

  $$I(1/100, 99/100) = 0.08 \text{ bits}$$

  Being told the result of the flip gives you much less information

Information content before asking about a feature

- Before testing a feature, two possible outcomes
  - example is positive with probability $p / (p + n)$
  - example is negative with probability $n / (p + n)$
- Information content of knowing the actual outcome

$$I\left(p / (p + n), n / (p + n) \right)$$

Information content after asking about a feature

- If we now test the feature *f*, how much information have we gained:

  Gain(*f*) =

  $$I\left(p / (p+n), n / (p+n) \right)$$

  $$- \sum_{i=1}^{k} (p_i + n_i) / (p+n)\, I\left(p_i / (p_i + n_i), n_i / (p_i + n_i) \right)$$

- Choose feature with largest information gain

Decision tree example

Suppose we have the following training data and we use the ID3 decision tree algorithm (with the information gain computations for selecting split variables) to induce a decision tree from the training data.

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

Root of the decision tree

When we begin, we have $p = 9$ positive and $n = 5$ negative examples in the training data. The amount of information contained in the correct answer is,

$$I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = I\left(\frac{9}{14}, \frac{5}{14}\right) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.940 \text{ bits.}$$

The possible attributes to split on are: Outlook, Temp, Humidity, and Wind. We now need to find the attribute that has the highest information gain; i.e., the attribute that gives us the most information about the correct classification. Recall that the general equation is,

$$\text{Gain(A)} = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) \; - \; \sum_{i=1}^{k} \frac{p_i + n_i}{p+n} \; I\left(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}\right)$$

where attribute A divides the examples into k subsets, and p_i and n_i represent the number of positive and negative examples in subset i, $i = 1, \ldots, k$.

Splitting on attribute Outlook would give:

Outlook = Sunny	p_1: 9, 11
	n_1: 1, 2, 8
Outlook = Overcast	p_2: 3, 7, 12, 13
	n_2:
Outlook = Rain	p_3: 4, 5, 10
	n_3: 6, 14

$$\text{Gain(Outlook)} = 0.940 - \left[\frac{5}{14}I\left(\frac{2}{5}, \frac{3}{5}\right) + \frac{4}{14}I\left(\frac{4}{4}, \frac{0}{4}\right) + \frac{5}{14}I\left(\frac{3}{5}, \frac{2}{5}\right)\right] = 0.247$$

Splitting on attribute Temp would give:

Temp = Hot	p_1: 3, 13
	n_1: 1, 2
Temp = Mild	p_2: 4, 10, 11, 12
	n_2: 8, 14
Temp = Cool	p_3: 5, 7, 9
	n_3: 6

$$\text{Gain(Temp)} = 0.940 - \left[\frac{4}{14} I\left(\frac{2}{4}, \frac{2}{4}\right) + \frac{6}{14} I\left(\frac{4}{6}, \frac{2}{6}\right) + \frac{4}{14} I\left(\frac{3}{4}, \frac{1}{4}\right)\right] = 0.029$$

Splitting on attribute Humidity would give:

Humidity = High	p_1: 3, 4, 12
	n_1: 1, 2, 8, 14
Humidity = Normal	p_2: 5, 7, 9, 10, 11, 13
	n_2: 6

$$\text{Gain(Humidity)} = 0.940 - \left[\frac{7}{14} I\left(\frac{3}{7}, \frac{4}{7}\right) + \frac{7}{14} I\left(\frac{6}{7}, \frac{1}{7}\right)\right] = 0.152$$

Splitting on attribute Wind would give:

Wind = Strong	p_1: 7, 11, 12
	n_1: 2, 6, 14
Wind = Weak	p_2: 3, 4, 5, 9, 10, 13
	n_2: 1, 8

$$\text{Gain(Wind)} = 0.940 - \left[\frac{6}{14} I\left(\frac{3}{6}, \frac{3}{6}\right) + \frac{8}{14} I\left(\frac{6}{8}, \frac{2}{8}\right)\right] = 0.048$$

So, Outlook will be the root of the decision tree. For Outlook = Sunny, there are still some positive and some negative instances, so we must repeat the procedure again along this branch (see below). For Outlook = Overcast, there is a leaf node, as all instances are positive. For Outlook = Rain, there are still some positive and some negative instances, so we must repeat the procedure again along this branch (see below).

Subtree rooted at Outlook = Sunny

For Outlook = Sunny, we first compute the amount of information contained in this subtree. There are 5 training examples, of which 2 are positive and 3 are negative:

p: 9, 11
n: 1, 2, 8

So the amount of information contained in this subtree is given by,

$$I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = I\left(\frac{2}{5}, \frac{3}{5}\right) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971$$

The possible attributes to split on are: Temp, Humidity, and Wind.

Splitting on attribute Temp would give:

Temp = Hot	p_1:
	n_1: 1, 2
Temp = Mild	p_2: 11
	n_2: 8
Temp = Cool	p_3: 9
	n_3:

$$\text{Gain(University)} = 0.971 - \left[\frac{2}{5}I\left(\frac{0}{2},\frac{2}{2}\right) + \frac{2}{5}I\left(\frac{1}{2},\frac{1}{2}\right) + \frac{1}{5}I\left(\frac{1}{1},\frac{0}{1}\right)\right] = 0.571$$

Splitting on attribute Humidity would give:

Humidity = High	p_1:
	n_1: 1, 2, 8
Humidity = Normal	p_2: 9, 11
	n_2:

$$\text{Gain(Humidity)} = 0.971 - \left[\frac{3}{5}I\left(\frac{0}{3},\frac{3}{3}\right) + \frac{2}{5}I\left(\frac{2}{2},\frac{0}{2}\right)\right] = 0.971$$

Splitting on attribute Wind would give:

Wind = Strong	p_1: 11
	n_1: 2
Wind = Weak	p_2: 9
	n_2: 1, 8

$$\text{Gain(Wind)} = 0.971 - \left[\frac{2}{5}I\left(\frac{1}{2},\frac{1}{2}\right) + \frac{3}{5}I\left(\frac{1}{3},\frac{2}{3}\right)\right] = 0.020$$

So, Humidity will be the root of the subtree. For Humidity = High there is a leaf node, as all instances are negative. For Humidity = Normal, there is a leaf node, as all instances are positive.

Subtree rooted at Outlook = Rain

For Outlook = Rain, we first compute the amount of information contained in this subtree. There are 5 training examples, of which 3 are positive and 2 are negative:

p: 4, 5, 10
n: 6, 14

So the amount of information contained in this subtree is given by,

$$I\left(\frac{p}{p+n},\frac{n}{p+n}\right) = I\left(\frac{3}{5},\frac{2}{5}\right) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971$$

The possible attributes to split on are: Temp, Humidity, and Wind.

Splitting on attribute Temp would give:

Temp = Hot	p_1:
	n_1:
Temp = Mild	p_2: 4, 10
	n_2: 14
Temp = Cool	p_3: 5
	n_3: 6

$$\text{Gain(Temp)} = 0.971 - \left[\frac{3}{5} I\left(\frac{2}{3}, \frac{1}{3}\right) + \frac{2}{5} I\left(\frac{1}{2}, \frac{1}{2}\right)\right] = 0.020$$

Splitting on attribute Humidity would give:

Humidity = High	p_1: 4
	n_1: 14
Humidity = Normal	p_2: 5, 10
	n_2: 6

$$\text{Gain(Humidity)} = 0.971 - \left[\frac{2}{5} I\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{3}{5} I\left(\frac{2}{3}, \frac{1}{3}\right)\right] = 0.020$$

Splitting on attribute Wind would give:

Wind = Strong	p_1:
	n_1: 6, 14
Wind = Weak	p_2: 4, 5, 10
	n_2:

$$\text{Gain(Wind)} = 0.971 - \left[\frac{2}{5} I\left(\frac{0}{2}, \frac{2}{2}\right) + \frac{3}{5} I\left(\frac{3}{3}, \frac{0}{3}\right)\right] = 0.971$$

So, Wind will be the root of the subtree. For Wind = Strong there is a leaf node, as all instances are negative. For Wind = Weak, there is a leaf node, as all instances are positive.

The final decision tree

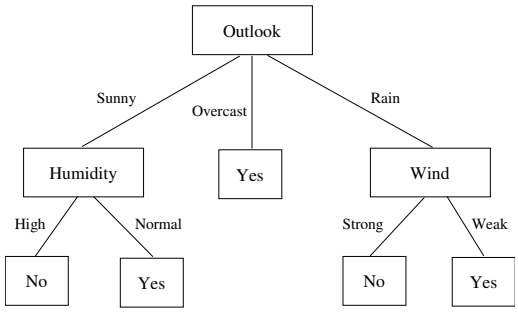


Figure 1: Final decision tree.