1 Chapter 18 Learning from Examples

1.1 Decision Trees

1.1.1 Introducing a decision tree

One of the simplest yet most successful forms of machine learning

Advantages of decision trees:

- Simple to understand and to interpret by a human.
- Performs well with a small data set
- Requires little data preparation.

Disadvantages of decision trees:

- Learning an optimal decision tree is NP-complete. Thus, a greedy heuristic approach is used in practice.
- The learning algorithm can create over-complex trees that do not generalize well.
- May not be able to represent some functions.
- Small variations in the data might result in a completely different tree being generated. (Use decision trees in conjunction with other learning algorithm.)

Take as input a vector of feature values and return a single output value.

For now: inputs have discrete values and the output has two possible values — a Binary classification

Each example input will be classified as true (positive example) or false (negative example).

We will use the following example to illustrate the decision tree learning algorithm.

Example: 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.

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.

Jeeves the valet – the 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 – the 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

A decision tree performs a sequence of tests in the input features.

- Each node performs a test on one input feature.
- Each arc is labeled with a value of the feature.
- Each leaf node specifies an output value.

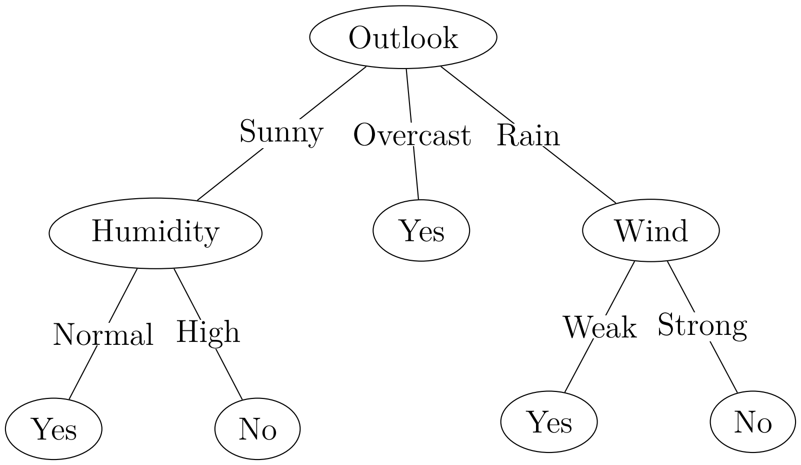
Using the Jeeves training set, we will construct two decision trees using different orders of testing the features.

Example 1: Let's construct a decision tree using the following order of testing features.

Test Outlook first.

For Outlook=Sunny, test Humidity. (After testing Outlook, we could test any of the three remaining features: Humidity, Wind, and Temp. We chose Humidity here.)

For Outlook=Rain, test Wind. (After testing Outlook, we could test any of the three remaining features: Humidity, Wind, and Temp. We chose Wind here.)



Example 2: Let's construct another decision tree by choosing Temp as the root node. This choice will result in a really complicated tree shown on the next page.

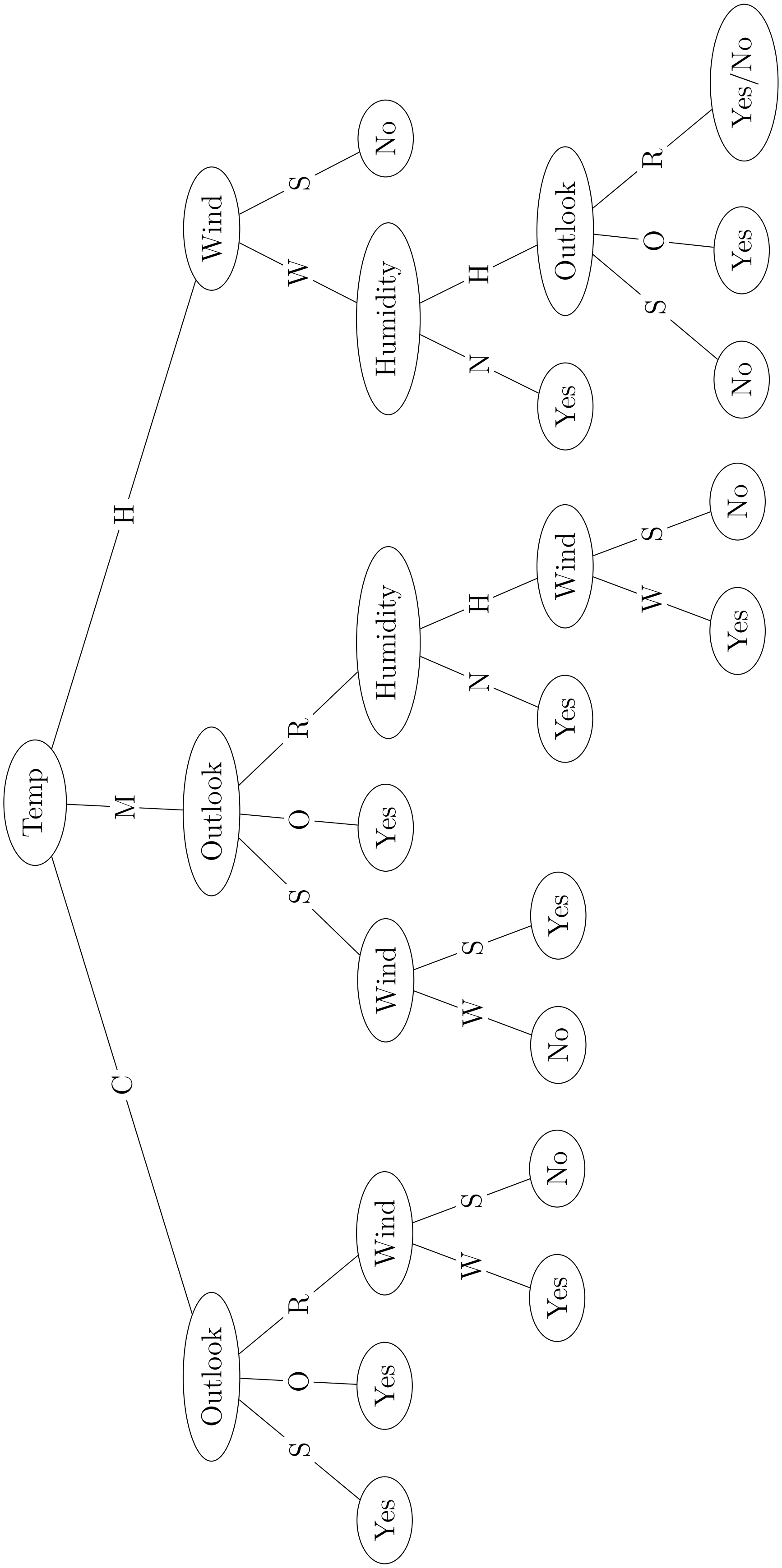
We have constructed two decision trees and both trees can classify the training examples perfectly. Which tree would you prefer?

One way to choose between the two is to evaluate them on the test set.

The first (and simpler) tree classifies 14/14 test examples correctly. Here are the decisions given by the first tree on the test examples. (1. No. 2. No. 3. No. 4. Yes. 5. Yes. 6. Yes. 7. Yes. 8. Yes. 9. Yes. 10. No. 11. Yes. 12. Yes. 13. No. 14. No.)

The second tree classifies 7/14 test examples correctly. Here are the decisions given by the second tree on the test examples. (1. Yes. 2. No. 3. No. 4. No. 5. Yes. 6. Yes/No. 7. Yes. 8. Yes. 9. Yes. 10. Yes. 11. Yes. 12. No. 13. Yes. 14. Yes.)

The second and more complicated tree performs worse on the test examples than the first tree, possibly because the second tree is overfitting to the training examples.


Temp
H
M
C
Wind
S
W
No
Humidity
H
N
Yes
Outlook
R
O
S
Yes/No
Yes
No
Outlook
R
O
S
Humidity
H
N
Wind
S
W
No
Yes
Yes
Yes
Wind
S
W
Yes
No
Outlook
R
O
S
Wind
S
W
No
Yes
Yes
Yes

Every decision tree corresponds to a propositional formula.

For example, our simpler decision tree corresponds to the propositional formula.

$(Outlook = Sunny \wedge Humidity = Normal) \vee (Outlook = Overcast) \vee (Outlook = Rain \wedge Wind = Weak)$

If we have n features, how many different functions can we encode with decisions trees? (Let's assume that every feature is binary.)

Each function corresponds to a truth table. Each truth table has 2^n rows. There are 2^{2^n} possible truth tables.

With $n = 10$, $2^{1024} \approx 10^{308}$

How do we find a good hypothesis in such a large space?

1.1.2 Constructing a decision tree

Want a tree that is consistent with the examples and is as small as possible.

Intractable to find the smallest consistent tree. (Intractable to search through $2^{2^{10}}$ function.

Use heuristics to find a small consistent tree.

The decision-tree-learning algorithm:

- A greedy divide-and-conquer approach
- Test the most important feature first.
- Solve the subproblems recursively.
- The most important feature makes the most difference to the classification of an example. We hope to minimize the number of tests to create a shallow tree.

The ID3 algorithm:

Algorithm 1 ID3 Algorithm (Features, Examples

```
If all examples are positive, return a leaf node with decision yes.
If all examples are negative, return a leaf node with decision no.
If no features left, return a leaf node with the most common decision of the examples.
If no examples left, return a leaf node with the most common decision of the examples in the parent.
else
    choose the most important feature $f$
    for each value $v$ of feature $f$ do
        add arc with label $v$
        add subtree $ID3(F - f, s \in S | f(s) = v)$
    end for
```

When would we encounter the base case "no features left"?

- We encounter this case when the data is noisy and there are multiple different decisions for the same set of feature values.
- See the following example.

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Weak	Yes
3	Sunny	Hot	High	Weak	Yes
4	Sunny	Hot	High	Weak	Yes

  These four data points all have the same feature values, but the decisions are different. This may happen if the decision is influenced by another feature that we don't observe. For example, the decision may be influenced by Bertie's mood when he woke up that morning, but Jeeves does not observe Bertie's mood directly.

- In this case, we return the majority decision of all the examples (breaking ties at random).

When would we encounter the base case "no examples left"?

- We encounter this base case when a certain combination of feature values does not appear in the training set.

  For example, the combination Temp = High, Wind = Weak, Humidity = High and Outlook = Rain does not appear in our training set.

- In this case, we will choose the majority decision among all the examples in the parent node (breaking ties at random).

For our example, the parent node of the right mode node is "Outlook". There are 2 examples at the node: 3 is a positive example and 1 is a negative example. We have a tie here. Thus, we will randomly choose one of the two decisions for this node (Thus, I labeled this node Yes/No.)

In the ID3 algorithm, the most important heuristic is choosing the most important feature to test at each step. How should we measure the importance of each feature and choose a feature that is the most important?

Intuitively, we want to choose a feature that allows us to make a decision as soon as possible. By doing this, we minimize the depth of the tree and keep the tree small.

We can measure the information content of each feature by comparing our uncertainty before and after testing the feature.

Suppose that before testing a feature, we have p positive examples and n negative examples. Let the feature have k values $v_1, \dots, v_k$. After testing the feature, for each value v_i, we will have p_i positive examples and n_i negative examples.

We can measure our uncertainty before and after testing a feature by the notion of "entropy", which comes from information theory.

Given a probability distribution $P(c_1), \dots, P(c_k)$ over k outcomes $c_1, \dots, c_k$. The entropy of the distribution is given by the formula below.

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

CQ:

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

  $I(0.5, 0.5) = -1/2log_2(1/2) - 1/2log_2(1/2) = 1.$

  There is one bit of uncertainty in this distribution.

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

  $I(0.99, 0.01) = -0.99log_2(0.99) - 0.01log_2(0.01) = 0.08.$

  There is 0.08 bit of uncertainty in this distribution.

  There is very little uncertainty in this distribution. We almost know for sure that the outcome will be the first one.

For a distribution over two outcomes, the entropy is maximized at $p = 1/2$ and is minimized at $p = 0$ and $p = 1$. By definition, $I(1, 0) = 0$ and $I(0, 1) = 0$.

Before testing a feature, there are two possible outcomes: The example is positive with probability $\frac{p}{p+n}$. The example is negative with probability $\frac{n}{p+n}$. Thus, the entropy before testing a feature is

$$I\left(\frac{p}{p+n}, \frac{n}{p+n}\right).$$

After testing a feature, the expected entropy is given by:

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

$\frac{p_i+n_i}{p+n}$ is the probability that the feature takes the value i, and $I\left(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}\right)$ is the entropy given that the feature takes the value i.

After testing a feature, the entropy should be reduced. Thus, the expected information gain by testing a feature is given by the entropy before testing the feature minus the expected entropy after testing the feature.

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

We will choose the feature with the largest information gain — This is the most important feature since it does the best job at reducing our uncertainty.

Applying the ID3 algorithm

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

When we begin, we have 9 positive and 5 negative examples in the training set.

Positive examples: 3, 4, 5, 7, 9, 10, 11, 12, 13 (9 examples) Negative examples: 1, 2, 6, 8, 14 (5 examples)

The entropy in the training set is

$$I\left(\frac{9}{14},\frac{5}{14}\right) = -\frac{9}{14} * log_2\left(\frac{9}{14}\right) - \frac{5}{14} * log_2\left(\frac{5}{14}\right) = 0.940.$$

The possible features to split on are: Outlook, Temp, Humidity, and Wind. We need to find the feature that has the highest information gain: i.e. the feature that gives us the largest reduction in the uncertainty of the data. Recall that the formula for calculating information gain is as follows.

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

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

Choosing the feature in the root node

If we split on Outlook, we would get

- Outlook = Sunny. +: 9, 11. -: 1, 2, 8.
- Outlook = Overcast. +: 3, 7, 12, 13. -: none.

- Outlook = Rain. +: 4, 5, 10. -: 6, 14.

$$
\begin{aligned}
&Gain(Outlook) && (1)\\
&= 0.940 - \left(\frac{5}{14} I\left(\frac{2}{5}, \frac{3}{5}\right) + \frac{4}{14} I\left(\frac{4}{4}, \frac{0}{4}\right) + \frac{5}{14} I\left(\frac{3}{5}, \frac{2}{5}\right)\right) && (2)\\
&= 0.940 - \left(\frac{5}{14} 0.971 + \frac{4}{14} 0 + \frac{5}{14} 0.971\right) && (3)\\
&= 0.940 - 0.694 && (4)\\
&= 0.247. && (5)
\end{aligned}
$$

If we split on Temp, we would get

- Temp = Cool. +: 5, 7, 9. -: 6.
- Temp = Mild. +: 4, 10, 11, 12. -: 8, 14.
- Temp = Hot. +: 3, 13. -: 1, 2.

$$
\begin{aligned}
&Gain(Temp) && (6)\\
&= 0.940 - \left(\frac{4}{14} I\left(\frac{2}{4}, \frac{2}{4}\right) + \frac{6}{14} I\left(\frac{4}{6}, \frac{2}{6}\right) + \frac{4}{14} I\left(\frac{3}{4}, \frac{1}{4}\right)\right) && (7)\\
&= 0.940 - \left(\frac{4}{14} 1 + \frac{6}{14} 0.918 + \frac{5}{14} 0.811\right) && (8)\\
&= 0.940 - 0.911 && (9)\\
&= 0.029. && (10)
\end{aligned}
$$

If we split on Wind, we would get

- Wind = Weak. +: 3, 4, 5, 9, 10, 13. -: 1, 8.
- Wind = Strong. +: 7, 11, 12. -: 2, 6, 14.

$$
\begin{aligned}
&Gain(Wind) && (11)\\
&= 0.940 - \left(\frac{6}{14} I\left(\frac{3}{6}, \frac{3}{6}\right) + \frac{8}{14} I\left(\frac{6}{8}, \frac{2}{8}\right)\right) && (12)\\
&= 0.940 - \left(\frac{6}{14}1 + \frac{8}{14}0.811\right) && (13)\\
&= 0.940 - 0.892 && (14)\\
&= 0.048. && (15)
\end{aligned}
$$

If we split on Humidity, we would get

- Humidity = Normal. +: 5, 7, 9, 10, 11, 13. -: 6.
- Humidity = High. +: 3, 4, 12. -: 1, 2, 8, 14.

$$
\begin{aligned}
&Gain(Humidity) && (16)\\
&= 0.940 - \left(\frac{7}{14} I\left(\frac{3}{7}, \frac{4}{7}\right) + \frac{7}{14} I\left(\frac{6}{7}, \frac{1}{7}\right)\right) && (17)\\
&= 0.940 - \left(\frac{7}{14}0.985 + \frac{7}{14}0.592\right) && (18)\\
&= 0.940 - 0.789 && (19)\\
&= 0.151 && (20)
\end{aligned}
$$

The expected information gain is the largest if we test Outlook. Thus, we will choose Outlook as the root of the decision tree.

- For Outlook = Sunny, there are both positive and negative examples. Thus, we need to repeat the procedure to choose a feature to test.
- For Outlook = Overcast, all examples are positive. So we add a leaf node with the decision Yes.
- For Outlook = Rain, there are both positive and negative examples. Thus, we need to repeat the procedure to choose a feature to test.

Subtree rooted at Outlook = Sunny

For Outlook = Sunny, we first compute the entropy of this subtree. There are 5 training examples, of which 2 are positive and 3 are negative. (+: 9, 11. -: 1, 2, 8.)

The entropy of this subtree is

$$\begin{aligned} & I\left(\frac{2}{5}, \frac{3}{5}\right) && (21) \\ & = -\frac{2}{5} log_2 \left(\frac{2}{5}\right) - \frac{3}{5} log_2 \left(\frac{3}{5}\right) && (22) \\ & = 0.971 && (23) \end{aligned}$$

The possible features to test are: Temp, Humidity, and Wind.

If we test Temp, we will get

- Temp = Cool. +: 9. -: none.
- Temp = Mild. +: 11. -: 8.
- Temp = Hot. +: none. -: 1, 2.

$$\begin{aligned} & Gain(Temp) && (24) \\ & = 0.971 - \left(\frac{2}{5} I\left(\frac{0}{2}, \frac{2}{2}\right) + \frac{2}{5} I\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{1}{5} I\left(\frac{1}{1}, \frac{0}{1}\right)\right) && (25) \\ & = 0.971 - \left(\frac{2}{5} 0 + \frac{2}{5} 1 + \frac{1}{5} 0\right) && (26) \\ & = 0.971 - 0.4 && (27) \\ & = 0.571. && (28) \end{aligned}$$

If we test Humidity, we will get

- Humidity = Normal. +: 9, 11. -: none.
- Humidity = High. +: none. -: 1, 2, 8.

$$
\begin{aligned}
&Gain(Humidity) && (29)\\
&= 0.971 - \left(\frac{3}{5}I\left(\frac{0}{3},\frac{3}{3}\right) + \frac{2}{5}I\left(\frac{2}{2},\frac{0}{2}\right)\right) && (30)\\
&= 0.971 - \left(\frac{3}{5}0 + \frac{2}{5}0\right) && (31)\\
&= 0.971 - 0 && (32)\\
&= 0.971 && (33)
\end{aligned}
$$

If we test Wind, we will get

- Wind = Weak. +: 9. -: 1, 8.
- Wind = Strong. +: 11. -: 2.

$$
\begin{aligned}
&Gain(Wind) && (34)\\
&= 0.971 - \left(\frac{2}{5}I\left(\frac{1}{2},\frac{1}{2}\right) + \frac{3}{5}I\left(\frac{1}{3},\frac{2}{3}\right)\right) && (35)\\
&= 0.971 - \left(\frac{2}{5}1 + \frac{3}{5}0.918\right) && (36)\\
&= 0.971 - 0.951 && (37)\\
&= 0.020. && (38)
\end{aligned}
$$

Thus, we will test Humidity at this node. For Humidity = Normal, we have a leaf node with the decision Yes since all examples are positive. For Humidity = High, we have a leaf node with the decision No since all examples are negative.

Subtree rooted at Outlook = Rain

For Outlook = Rain, we first compute the entropy of this subtree. There are 5 training examples, of which 3 are positive and 2 are negative. (+: 4, 5, 10. -: 6, 14.)

The entropy of this subtree is

$$\begin{aligned}
& I\left(\frac{3}{5}, \frac{2}{5}\right) && (39)\\
& = -\frac{3}{5} log_2\left(\frac{3}{5}\right) - \frac{2}{5} log_2\left(\frac{2}{5}\right) && (40)\\
& = 0.971 && (41)
\end{aligned}$$

The possible features to test are: Temp, Humidity, and Wind.

If we test Temp, we will get

- Temp = Cool. +: 5. -: 6.
- Temp = Mild. +: 4, 10. -: 14.
- Temp = Hot. +: none. -: none.

$$\begin{aligned}
& Gain(Temp) && (42)\\
& = 0.971 - \left(\frac{3}{5} I\left(\frac{2}{3}, \frac{1}{3}\right) + \frac{2}{5} I\left(\frac{1}{2}, \frac{1}{2}\right)\right) && (43)\\
& = 0.971 - \left(\frac{3}{5} 0.918 + \frac{2}{5} 1\right) && (44)\\
& = 0.971 - 0.951 && (45)\\
& = 0.02. && (46)
\end{aligned}$$

If we test Humidity, we will get

- Humidity = Normal. +: 5, 10. -: 6.
- Humidity = High. +: 4. -: 14.

$$\begin{aligned}
&Gain(Humidity) && (47)\\
&= 0.971 - \left(\frac{3}{5}I\left(\frac{2}{3},\frac{1}{3}\right) + \frac{2}{5}I\left(\frac{1}{2},\frac{1}{2}\right)\right) && (48)\\
&= 0.971 - \left(\frac{3}{5}0.918 + \frac{2}{5}1\right) && (49)\\
&= 0.971 - 0951 && (50)\\
&= 0.02 && (51)
\end{aligned}$$

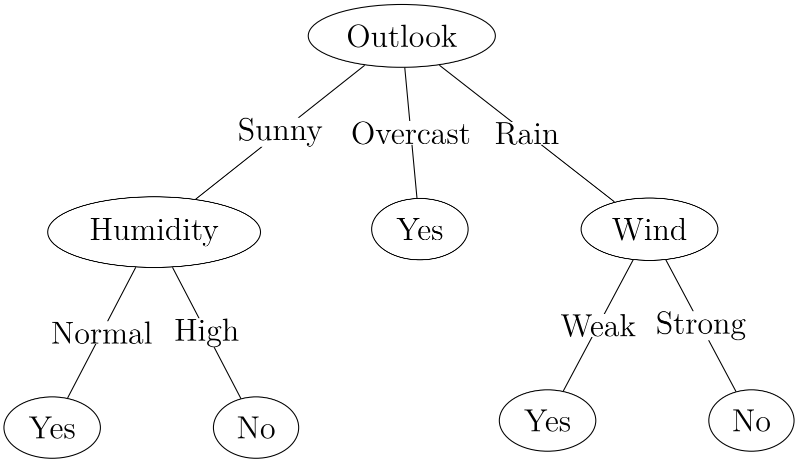
If we test Wind, we will get

- Wind = Weak. +: 4, 5, 10.. -: none.
- Wind = Strong. +: none. -: 6, 14.

$$\begin{aligned}
&Gain(Wind) && (52)\\
&= 0.971 - \left(\frac{2}{5}I\left(\frac{0}{2},\frac{2}{2}\right) + \frac{3}{5}I\left(\frac{3}{3},\frac{0}{3}\right)\right) && (53)\\
&= 0.971 - \left(\frac{2}{5}0 + \frac{3}{5}0\right) && (54)\\
&= 0.971 - 0 && (55)\\
&= 0.971. && (56)
\end{aligned}$$

Thus, we will test Wind at this node. For Wind = Weak, we have a leaf node with the decision Yes since all examples are positive. For Wind = Strong, we have a leaf node with the decision No since all examples are negative.

The final decision tree is, unsurprisingly, the first one we've seen.


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