Dat550-exercise2 Solutions

Vinay Setty

30. January 2020

1 Entropy

Consider the following dataset with 8 documents d_1 to d_8 and features/attributes f_1 to f_3.

Doc	f_1	f_2	f_3	Class
d1	2	0	0	c_1(Algebra)
d2	2	0	0	c_1(Algebra)
d3	0	0	0	c_2(Calculus)
d4	0	1	0	c_2(Calculus)
d5	0	2	0	c_3(Stochastics)
d6	0	2	0	c_3(Stochastics)
d7	0	1	1	c_3(Stochastics)
d8	0	2	1	c_3(Stochastics)

1. What is the entropy over the categories for these training instances d1 to d8? Recall that the entropy of a partition $\mathcal{T}$ is given as $H(\mathcal{T}) = -\sum_j \mathrm{P}(\mathcal{T}_j) \cdot \log_2 \mathrm{P}(\mathcal{T}_j)$. Note the log_2 has been chosen to make the calculations simple for you. You actually do not need a calculator.

2. Using the training set d1 to d8, suppose we want to construct a decision tree for the binary classification of the category c3 ("Stochastics"), i.e., the tree decides whether a new document belongs to c3 category or not, using binary splits. Determine the split with the highest information gain for binary split at the root level. Recall Information gain formula

$$G(k, k_1, k_2) = H(k) - \frac{|k_1|}{|k|} H(k_1) - \frac{|k_2|}{|k|} H(k_2)$$

$f_1 \geq 1$ $f_2 \geq 1$ $f_3 \geq 1$

$f_1 \geq 2$ $f_2 \geq 2$ $f_3 \geq 2$

$f_1 \geq 3$ $f_2 \geq 3$ $f_3 \geq 3$

Solution

1. $$H(\mathcal{T}) = -1 \cdot \left(-\frac{2}{8}\log_2\frac{2}{8} - \frac{2}{8}\log_2\frac{2}{8} - \frac{4}{8}\log_2\frac{4}{8} \right) H(\mathcal{T}) = +0.5{+}0.5{+}0.5 = 1.5$$

2. $f_2 \geq 1$

2 Build a Decision Tree

Construct a decision tree given the following training data set.

Table 1: Data

Outlook	**Temp.**	**Humidity**	**Windy**	**Play**
sunny	85	85	false	No
sunny	80	90	true	No
overcast	83	78	false	Yes
rain	70	96	false	Yes
rain	68	80	false	Yes
rain	65	70	false	Yes
overcast	64	65	true	Yes
sunny	72	95	false	No
sunny	69	70	false	Yes
rain	75	80	false	Yes
sunny	75	70	true	Yes
overcast	72	90	true	Yes
overcast	81	75	false	Yes
rain	71	80	true	No

Solution

We call the algorithm with all attributes and the entire training set.

`[1]: ID3(R={Outlook, Temperature, Humidity, Windy}, C=Play, S={1,…,14})`

(We refer to the instances in the training set by the row numbers of the records).

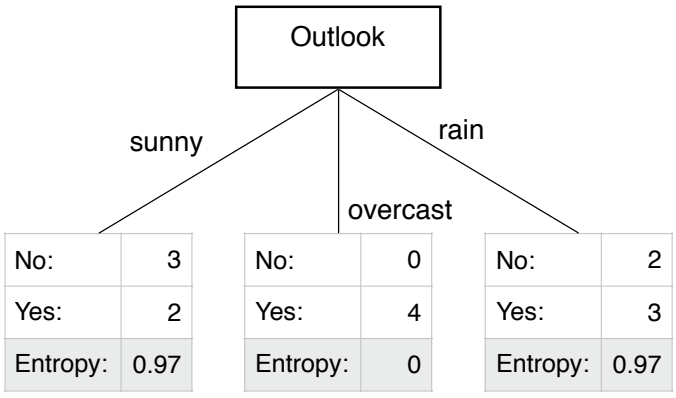
None of the if conditions in first three bullet points is true, so we need to select the attribute with the highest gain. I.e., we need to compute Gain for all 4 attributes.

Notes:

- The initial entropy is computed on the entire training set:

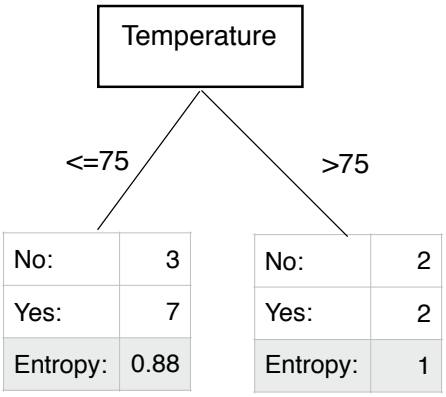
No:	5
Yes:	9
Entropy:	0.94

- We split the continuous attributes (Temperature and Humidity) arbitrarily to two categories based on the values (<=75 and >75).
- We only consider the training instances at the children nodes (when counting Yes/No-s) that have the specific attribute value.



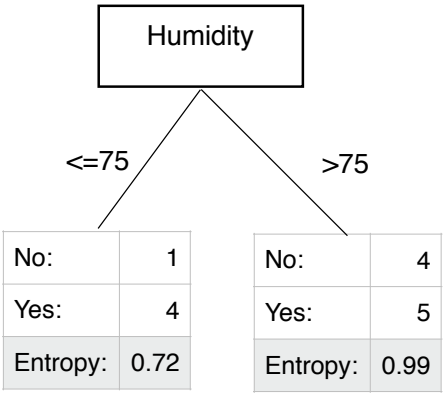
No:	3	No:	0	No:	2
Yes:	2	Yes:	4	Yes:	3
Entropy:	0.97	Entropy:	0	Entropy:	0.97

Gain: 0.246



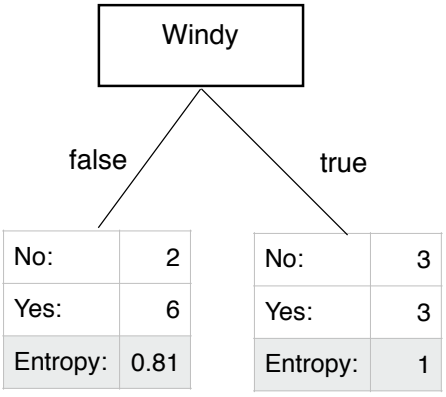
No:	3	No:	2
Yes:	7	Yes:	2
Entropy:	0.88	Entropy:	1

Gain: 0.024



No:	1	No:	4
Yes:	4	Yes:	5
Entropy:	0.72	Entropy:	0.99

Gain: 0.045



No:	2	No:	3
Yes:	6	Yes:	3
Entropy:	0.81	Entropy:	1

Gain: 0.047

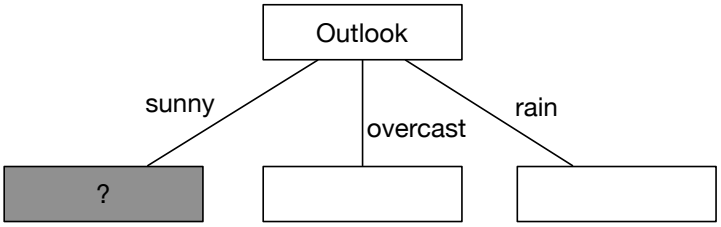
We find that *Outlook* has the highest gain, so we'll first split on this attribute.

We need to call ID3 recursively on each resulting node.

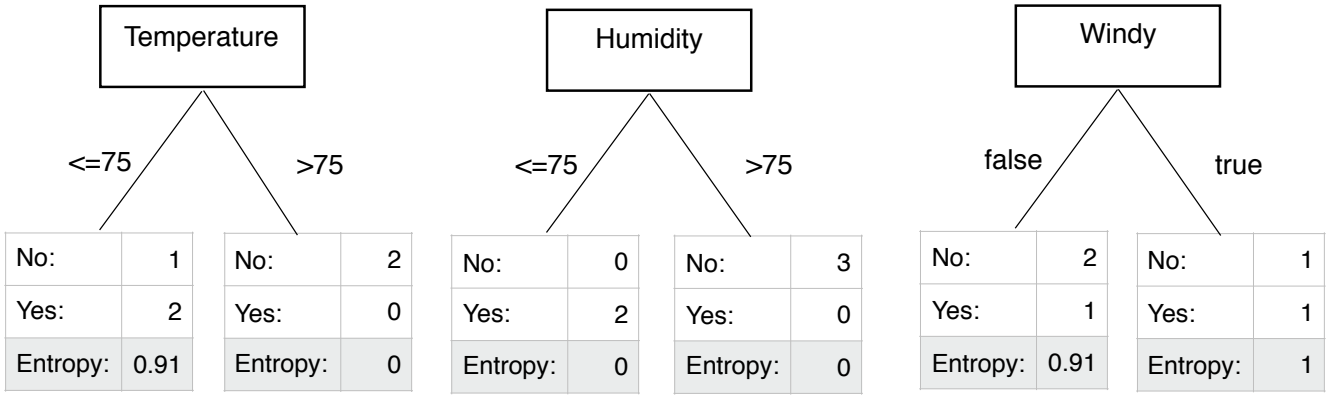
- The attribute Outlook has to be removed from the set of attributes considered (R).
- For each node we only consider the training records where Outlook has the corresponding attribute value (sunny, rain, or overcast).

Let's look at the three recursive calls (2.1, 2.2, 2.3) one by one.

`[2.1]: ID3(R={Temperature, Humidity, Windy}, C=Play, S={1,2,8,9,11})`



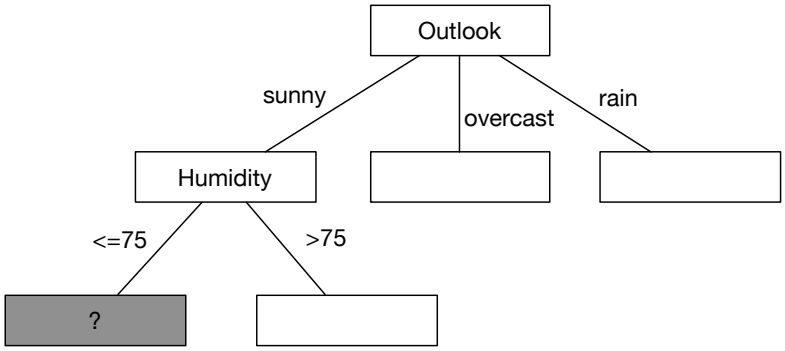
None of the if conditions in first three bullet points is true, so we need to compute gain for each of the tree attributes (Temperature, Humidity, Windy). But, when computing the instances belonging to the No/Yes classes, we only consider those where Outlook=sunny.
The parent entropy is the entropy we computed for the "sunny" node: 0.97.



Humidity has the highest gain so we'll split on this attribute.

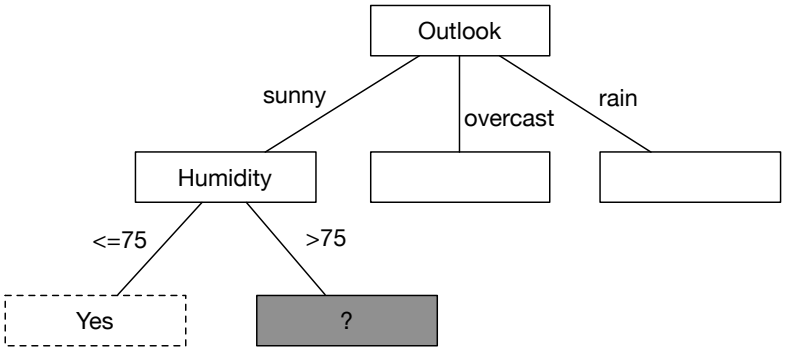
We need to call the algorithm recursively on both nodes (<=75 and >75).

`[2.1.1] ID3(R={Temperature, Windy}, C=Play, S={9,11})`



Running the algorithm we find that in that all records have the same target value (Yes). This means that the condition in the second bullet point is met; **this will be a leaf node with value Yes**.

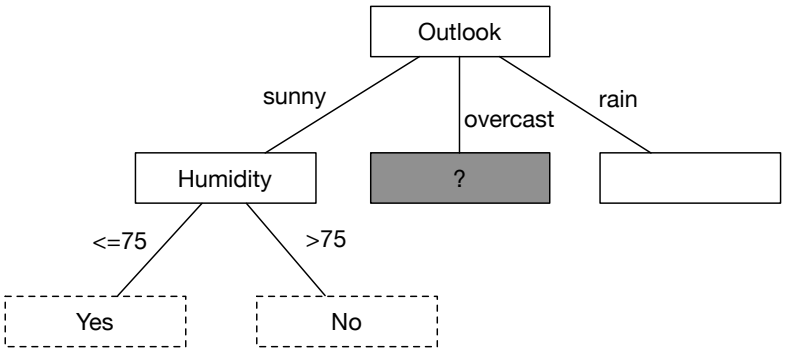
`[2.1.2] ID3(R={Temperature, Windy}, C=Play, S={1,2,8})`



The situation for this node is exactly the same: all records have the same target value (No). **This will be a leaf node with value No**.

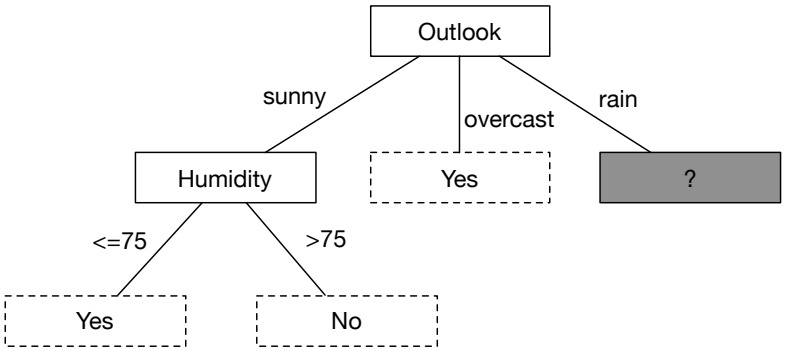
Now we need to recurse back one level, where the algorithm is called for Outlook=overcast.

`[2.2] ID3(R={Temperature, Humidity, Windy}, C=Play, S={3,7,12,13})`

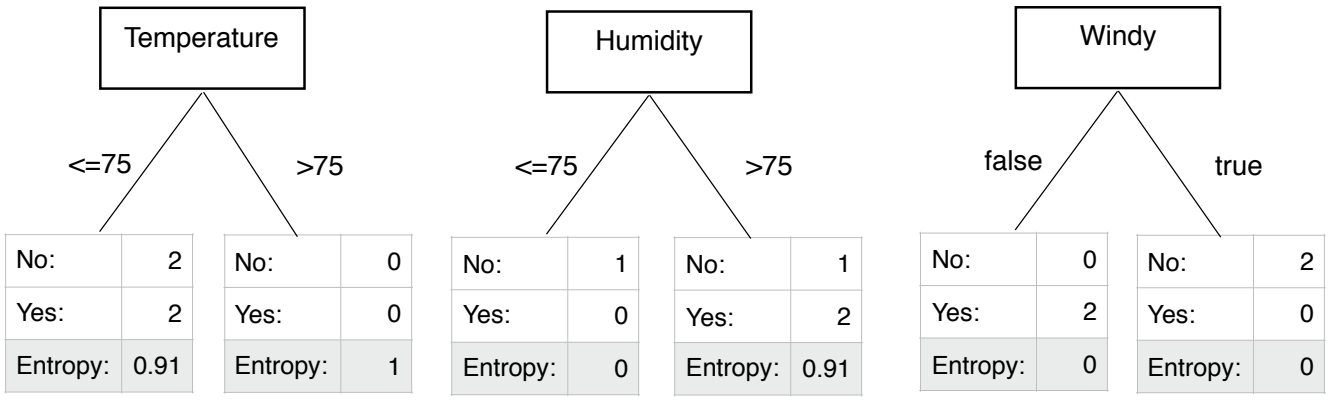


Here again all records have the same target value (Yes). **This will be a leaf node with label Yes.** Next the algorithm is called for Outlook=rain.

[2.3] ID3(R={Temperature, Humidity, Windy}, C=Play, S={4,5,6,14})



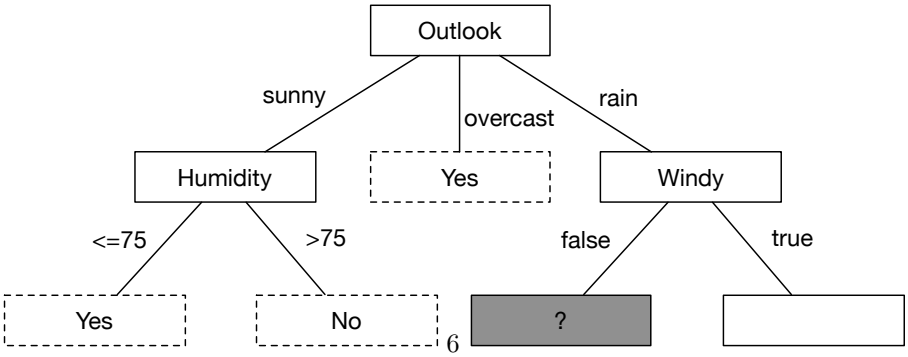
We find both Yeses and Nos as target values, therefore we proceed exactly as we did for "sunny".



Windy has the highest gain so we'll split on this attribute.

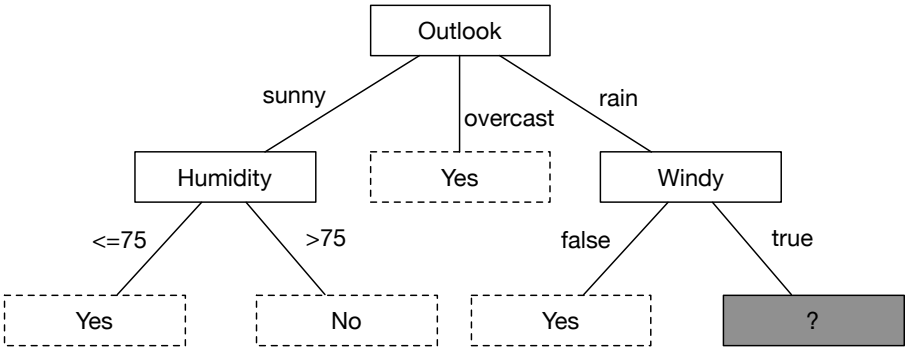
We need to call the algorithm recursively on both nodes (false, true).

[2.3.1] ID3(R={Temperature, Humidity}, C=Play, S={4,5})



All records have the same target value; **this becomes a leaf node with label Yes.**

[2.3.2] ID3(R={Temperature, Humidity}, C=Play, S={6,14})



All records have the same target value; **this becomes a leaf node with label No**.

There is nowhere to recurse anymore. This is the final decision tree:

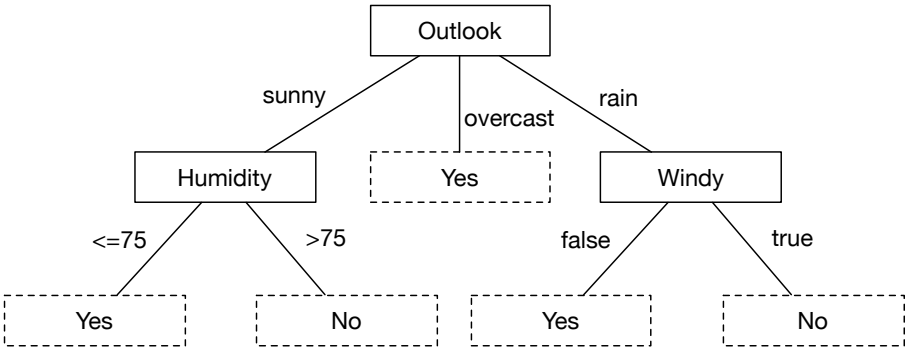


Table 2: Attributes

Outlook	categorical (sunny, overcast, rain)
Temperature	continuous (in Fahrenheit)
Humidity	continuous (percentage)
Windy	categorical (true, false)
Play	categorical class/label/target (Yes, No)

For the above task use the ID3 Algorithm
function ID3 (R: a set of attributes, C: the target attribute, S: a training set) returns a decision tree

- If S is empty, return a leaf node with the default class (majority class in the entire training set).
- If S consists of records all with the same value for the target attribute, return a single node with that value (this will be a leaf node).
- If R is empty, then return a single node with as value the most frequent of the values of the target attribute that are found in records of S (this will be a leaf node; note that then there will be errors, that is, records that will be improperly classified).
- Otherwise (if none of the previous conditions are met): Let D be the attribute with largest $Gain(D, S)$ among the attributes in R (You may also use gain ratio instead of gain).

$$Gain = Entropy(p) - \sum_{j=1}^{k} \frac{N(v_j)}{N} Entropy(v_j)$$

where k is the number of attribute values, N is the total number of records at the parent node ($= |S|$), $N(v_j)$ is the number of records associated with the child node v_j. The Entropy for two classes ($C = No, C = Yes$):

$$Entropy(t) = -P(C = No|t) \cdot \log_2 P(C = No|t) - P(C = Yes|t) \cdot \log_2 P(C = Yes|t)$$

- Let $\{d_j | j = 1, 2, ..., m\}$ be the values of attribute D. Let $\{S_j | j = 1, 2, ..., m\}$ be the subsets of S consisting respectively of records with value dj for attribute D.
- Return a tree with root labeled D and edges labeled $d_1, d_2, ..., d_m$ going respectively to the trees $ID3(R-D, C, S1), ID3(R-D, C, S2), ..., ID3(R-D, C, Sm)$