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).

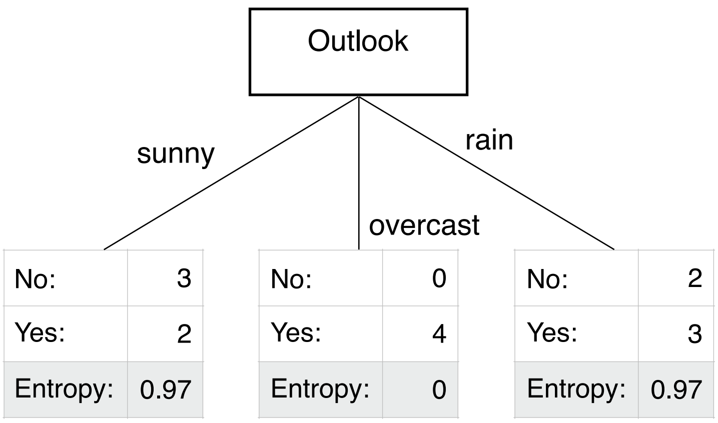
None of the if conditions in first three bullet points is true, so we need to select the attribute with the highest gain. I.e., we need to compute Gain for all 4 attributes.

Notes:

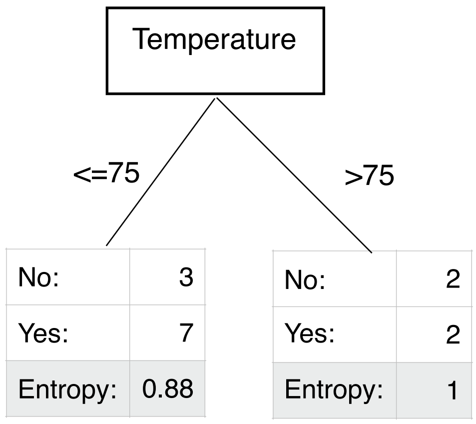
- The initial entropy is computed on the entire training set:

No:	5
Yes:	9
Entropy:	0.94

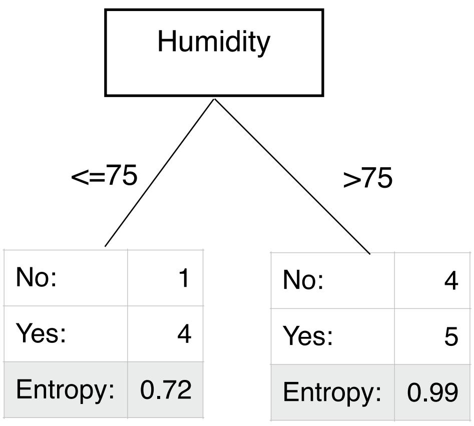
- We split the continuous attributes (Temperature and Humidity) arbitrarily to two categories based on the values (<=75 and >75).
- We only consider the training instances at the children nodes (when counting Yes/No-s) that have the specific attribute value.



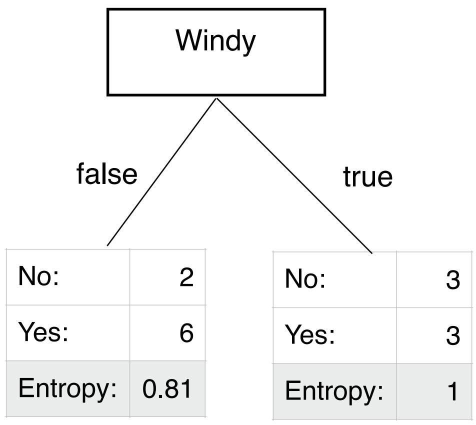
Gain: 0.246



Gain: 0.024



Gain: 0.045



Gain: 0.047

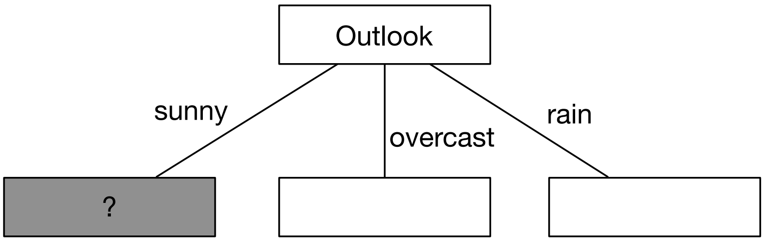
We find that *Outlook* has the highest gain, so we'll first split on this attribute.

We need to call ID3 recursively on each resulting node.

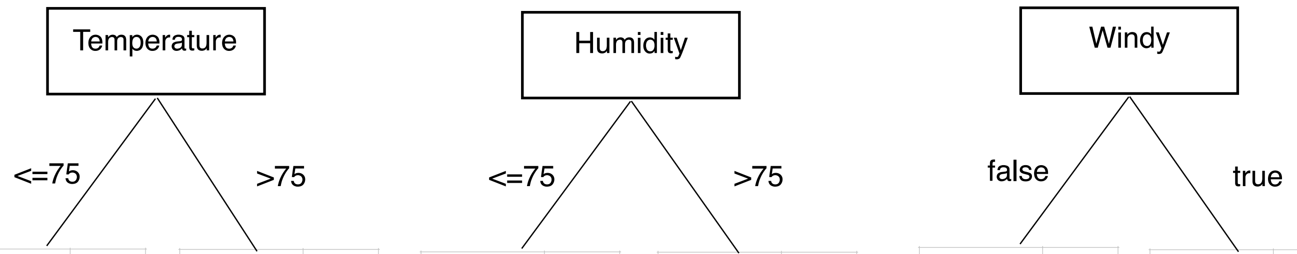
- The attribute Outlook has to be removed from the set of attributes considered (R).
- For each node we only consider the training records where Outlook has the corresponding attribute value (sunny, rain, or overcast).

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.



Temperature <=75:

No:	1
Yes:	2
Entropy:	0.91

Temperature >75:

No:	2
Yes:	0
Entropy:	0

Gain: 0.419

Humidity <=75:

No:	0
Yes:	2
Entropy:	0

Humidity >75:

No:	3
Yes:	0
Entropy:	0

Gain: 0.97

Windy false:

No:	2
Yes:	1
Entropy:	0.91

Windy true:

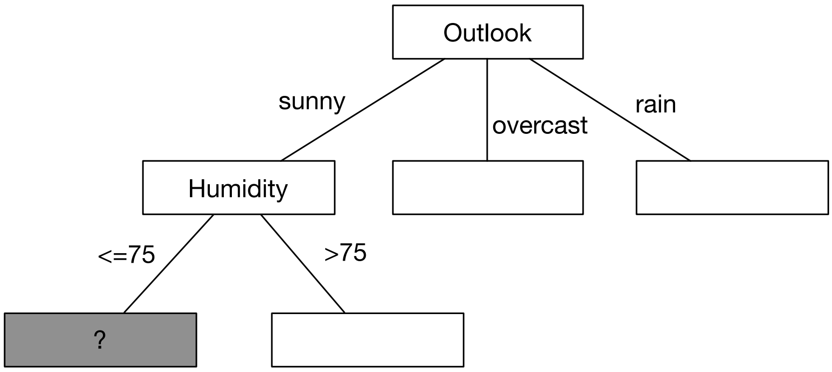
No:	1
Yes:	1
Entropy:	1

Gain: 0.019

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

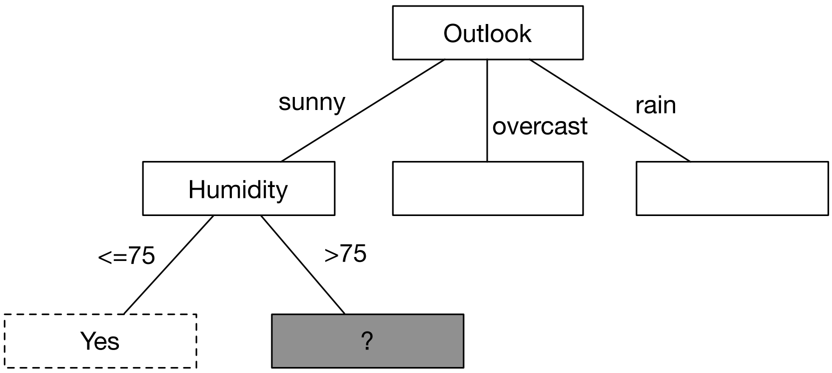
We need to call the algorithm recursively on both nodes (<=75 and >75).

```
[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**.

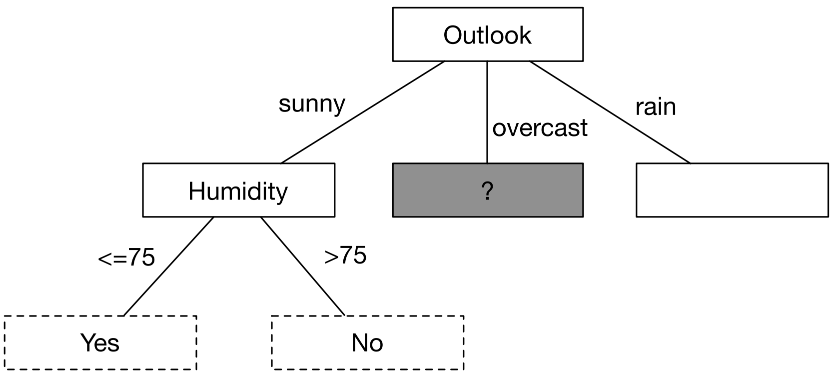
`[2.1.2] ID3(R={Temperature, Windy}, C=Play, S={1,2,8})`



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.

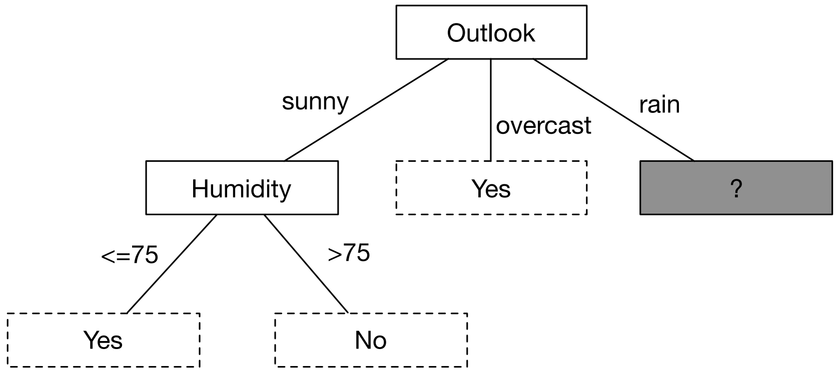
Now we need to recurse back one level, where the algorithm is called for Outlook=overcast.

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

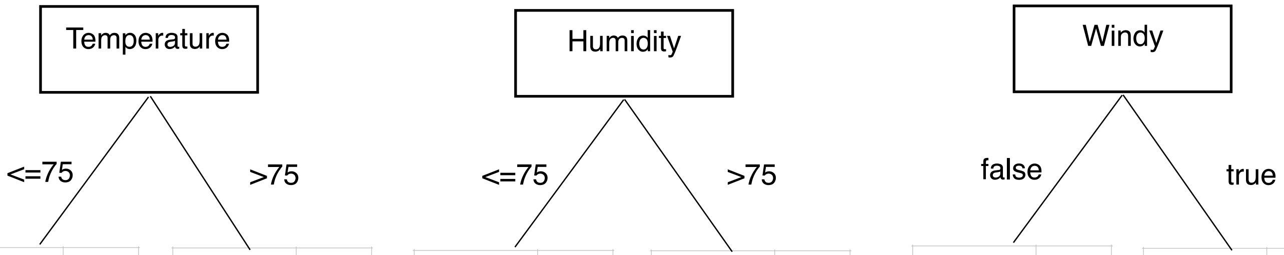


Here again all records have the same target value (Yes). **This will be a leaf node with label Yes.**
Next the algorithm is called for Outlook=rain.

```
[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".



Temperature <=75

No:	2
Yes:	2
Entropy:	0.91

Temperature >75

No:	0
Yes:	0
Entropy:	1

Gain: -0.03

Humidity <=75

No:	1
Yes:	0
Entropy:	0

Humidity >75

No:	1
Yes:	2
Entropy:	0.91

Gain: 0.28

Windy false

No:	0
Yes:	2
Entropy:	0

Windy true

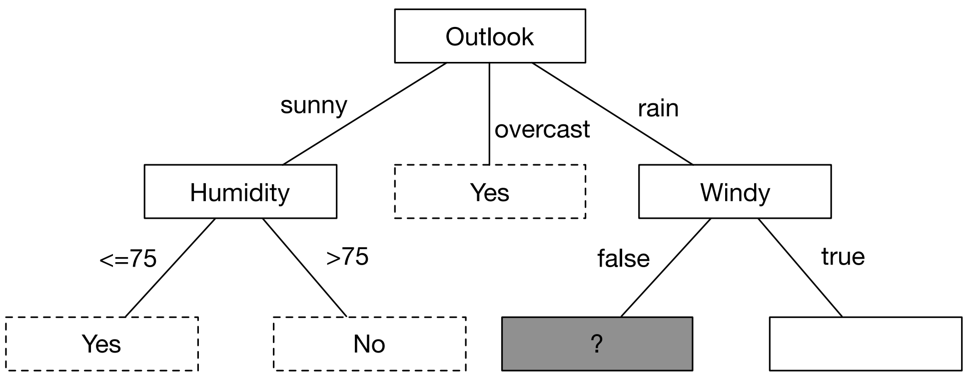
No:	2
Yes:	0
Entropy:	0

Gain: 0.97

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

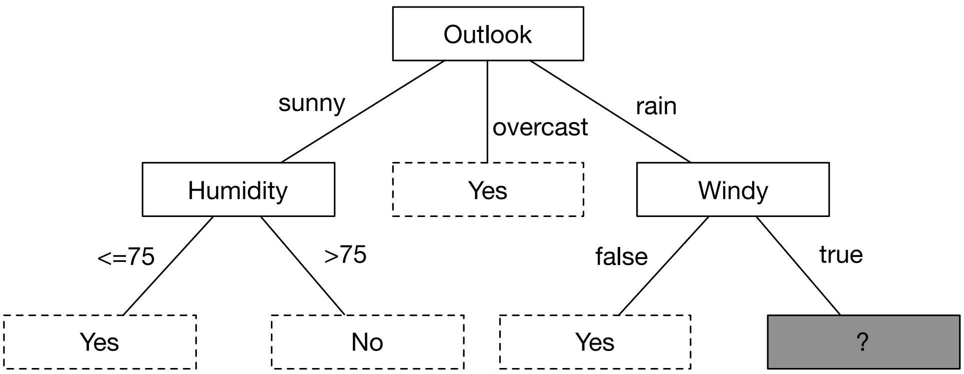
We need to call the algorithm recursively on both nodes (false, true).

```
[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: