Decision Tree Learning

Represents disjunction of conjunction of constraints. on
$\qquad$ (v) $\qquad\qquad$ (∧)
values attributes can get.

(Outlook = Sunny ∧ Humidity = Normal) v (Outlook = Overcast)
$\qquad\qquad$ v (Outlook = Rain ∧ Wind = Weak)

```
                    ┌─────────┐
                    │ Outlook │
                    └─────────┘
          Sunny    Overcast    Rain
         /           |            \
   ┌───────┐        Yes      ┌──────┐
   │ Humid │                 │ Wind │  Conjunction
   └───────┘                 └──────┘
   Normal  High  Disjunction  Weak  Strong
    /       \                  /       \
  Yes       No               yes       No
```
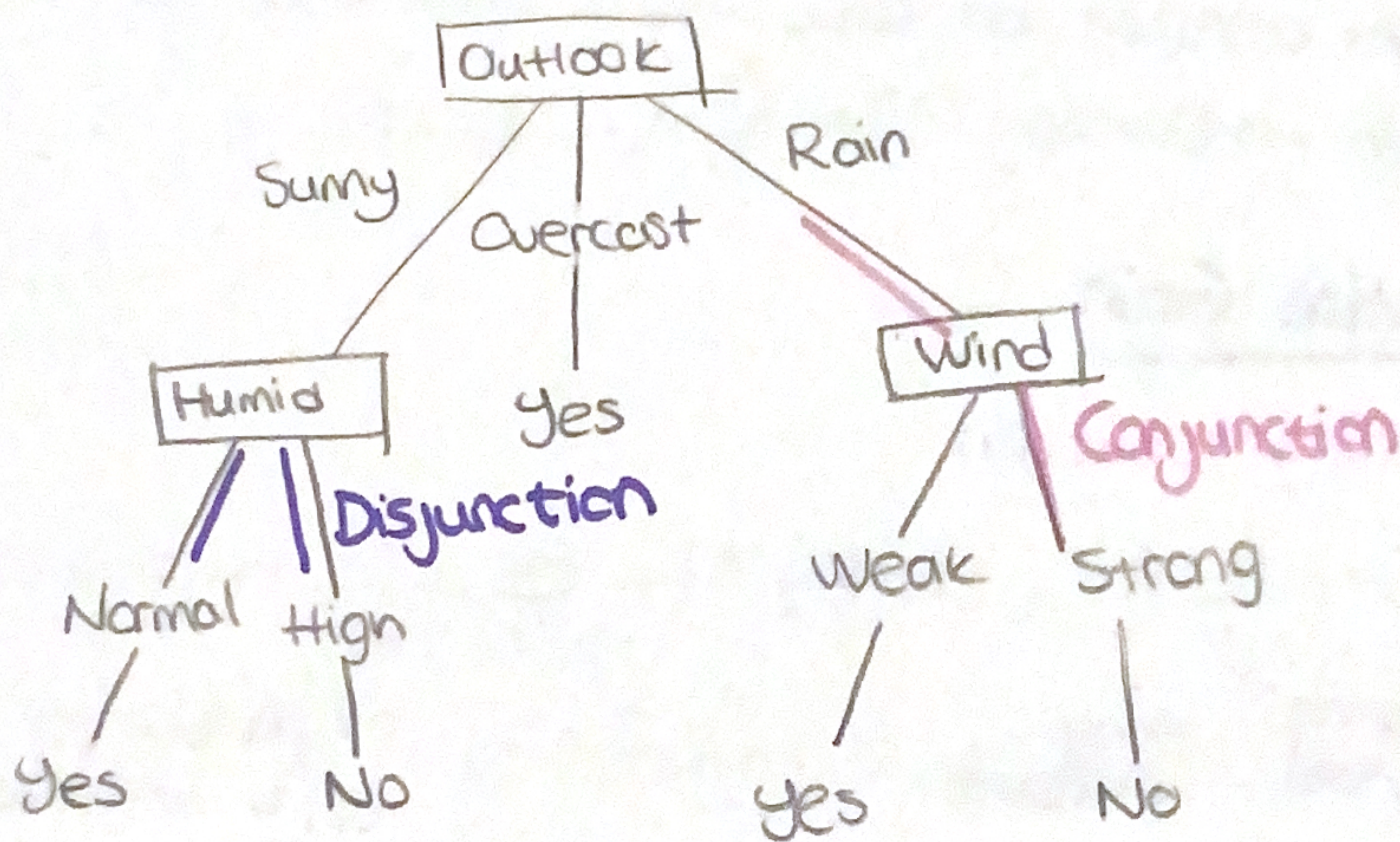
ID3
Algorithm
1. Create a root node for the tree
2. If all examples are (+) return single node tree [+]
3. Do the same if all examples are (−)
4. If attributes are empty return tree root with most common value of target attribute.

Otherwise begin

$\qquad\qquad\qquad\qquad\qquad$ Dataset's − Entropy
$\qquad\qquad\qquad\qquad\nearrow$ entropy $\quad$ of
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ attribute

1. Find attribute with best information gain = A
2. Put that attribute to root node (A)
3. For each possible value of A (vi) do:
    1. Add a new tree branch below root, A = vi
    2. If examples of A = vi empty, put the most common target label
    3. If examples of A = vi exist, add the subtree with ID3 {Examples$_{vi}$, Target attribute, Attributes − A}
$\qquad\qquad\qquad$ ↳ recursion