Decision Tree Algorithm

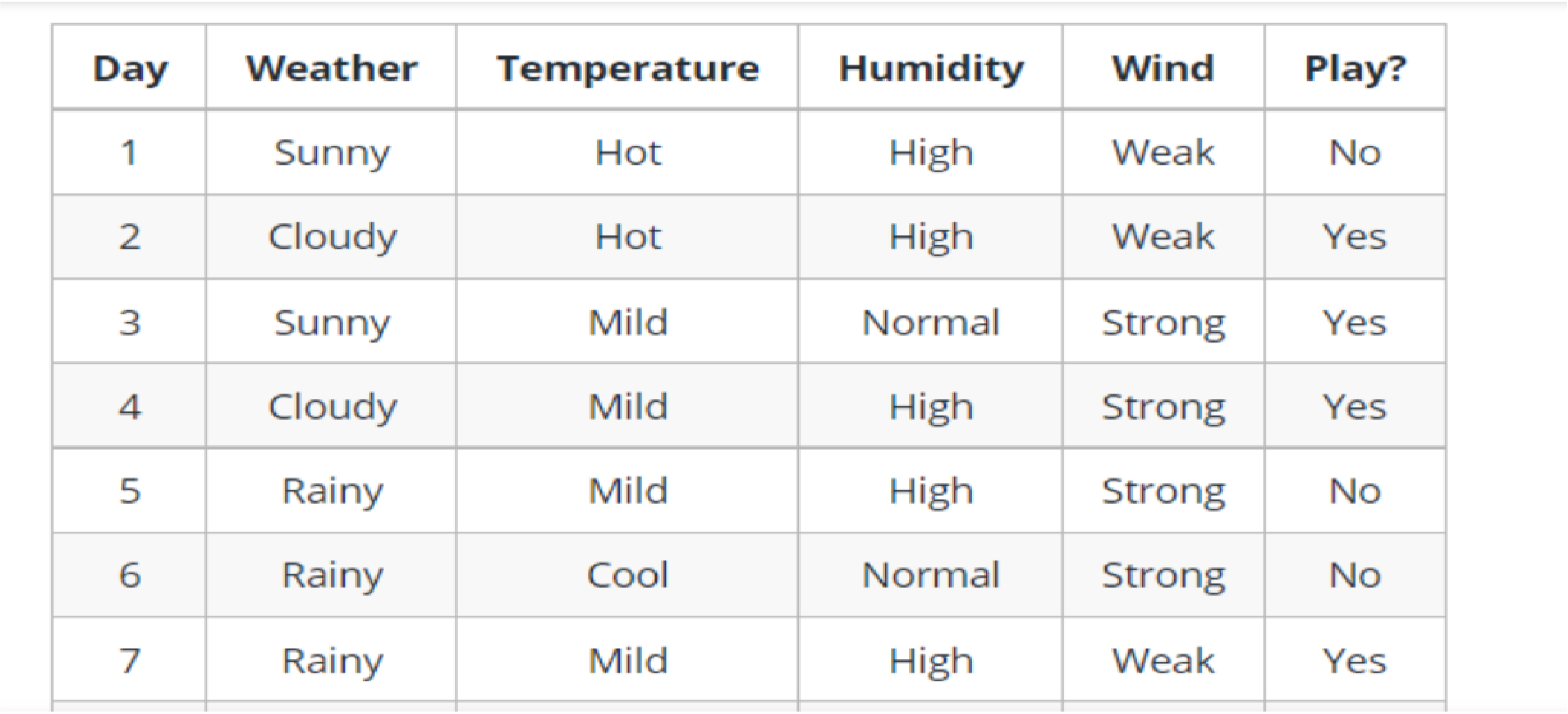
Lab4

Decision Tree

- Decision trees can be used for classification as well as regression problems.
- The name itself suggests that it uses a flowchart like a tree structure to show the predictions that result from a series of feature-based splits.

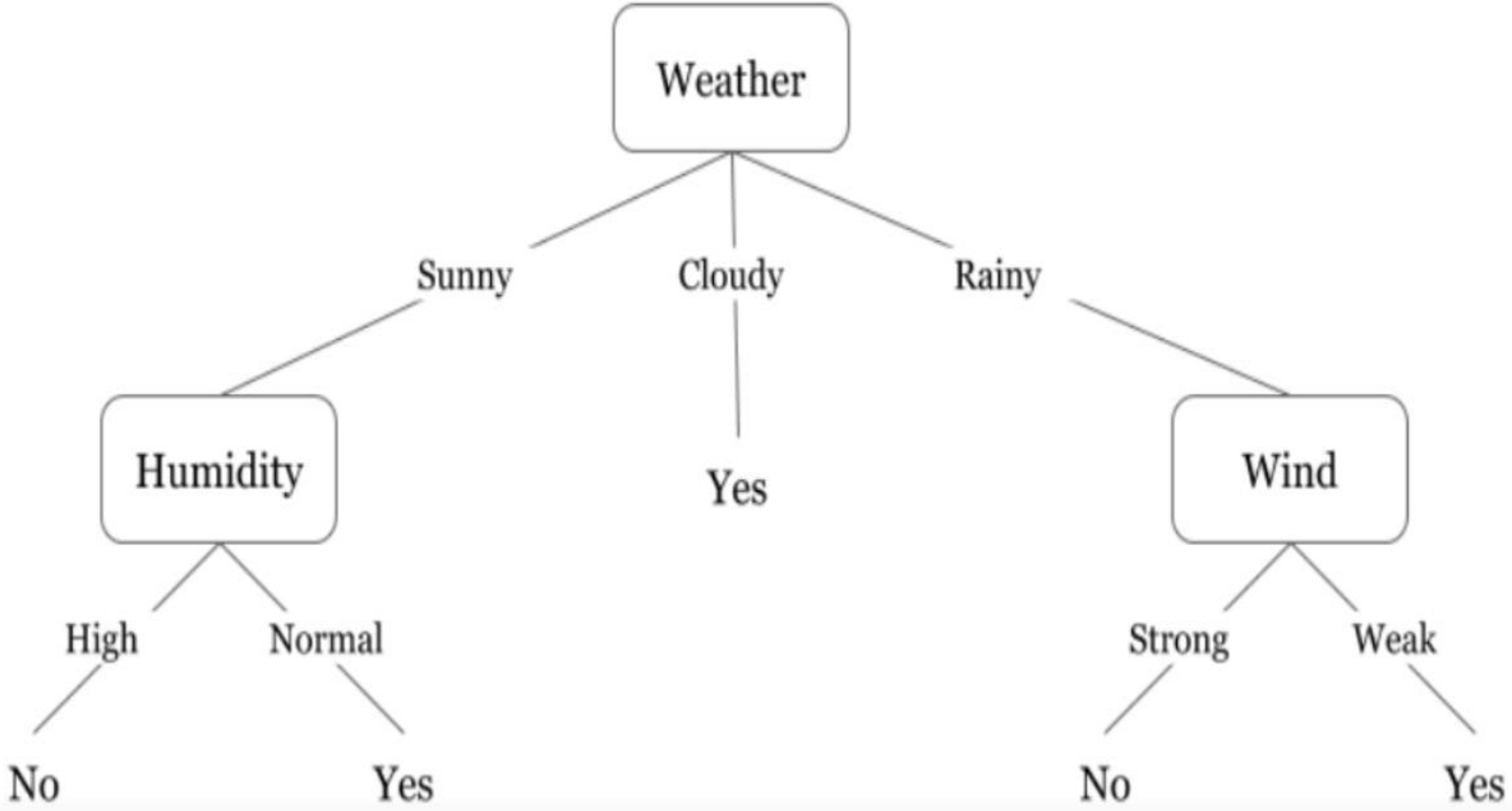
- ***Root Nodes*** – It is the node present at the beginning of a decision tree from this node the population starts dividing according to various features.
- ***Decision Nodes*** – the nodes we get after splitting the root nodes are called Decision Node
- ***Leaf Nodes*** – the nodes where further splitting is not possible are called leaf nodes or terminal nodes
- ***Sub-tree*** – just like a small portion of a graph is called sub-graph similarly a sub-section of this decision tree is called sub-tree.
- ***Pruning*** – is nothing but cutting down some nodes to stop overfitting.

Example



Day	Weather	Temperature	Humidity	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes

Example


Weather
Sunny
Cloudy
Rainy
Humidity
Yes
Wind
High
Normal
Strong
Weak
No
Yes
No
Yes

Entropy

Entropy is nothing but the uncertainty in our dataset or measure of impurity.

The formula for Entropy is shown below:

$$E(S) = -p_{(+)}\log p_{(+)} - p_{(-)}\log p_{(-)}$$

Here p_+ is the probability of positive class

p_- is the probability of negative class

S is the subset of the training example

Entropy

- The higher the Entropy, the lower will be the purity and the higher will be the impurity.
- Calculate the entropy of each node/ feature.
- Select that node for the split which has the least impurity or least entropy.

Information Gain

- Information gain measures the reduction of uncertainty given some feature and it is also a deciding factor for which attribute should be selected as a decision node or root node.

$$Information\ Gain\ =\ E(Y)\ -\ E(Y|X)$$

- We use information gain to decide which feature should be the root node and which feature should be placed after the split.
- We select the feature which has the highest information gain for the split.

When to stop splitting?

- ***max_depth*** parameter:The more the value the more complex your tree will be.
- ***min_samples_split***: Here we specify the minimum number of samples required to do a spilt
- ***min_samples_leaf*** :represents the minimum number of samples required to be in the leaf node. The more you increase the number, the more is the possibility of overfitting.
- ***max_features*** :It helps us decide what number of features to consider when looking for the best split.

Pruning

- It is another method that can help us avoid overfitting.
- It helps in improving the performance of the tree by cutting the nodes or sub-nodes which are not significant.
- It removes the branches which have very low importance.