



Lesson_10: Decision Trees

Ali Aburas, PhD

Outline



1. Decision Trees
2. Entropy/Information Gain
3. ID3 Algorithm

Decision Tree Classification Algorithm

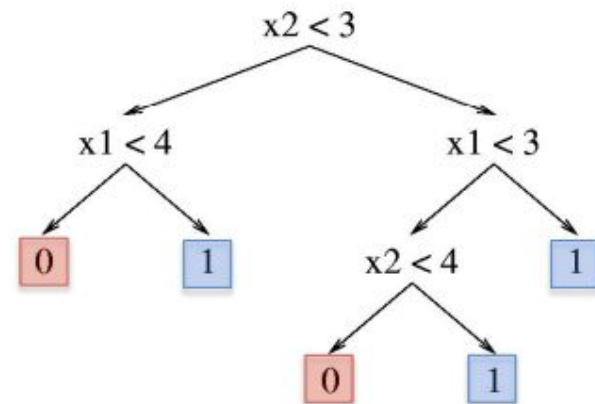
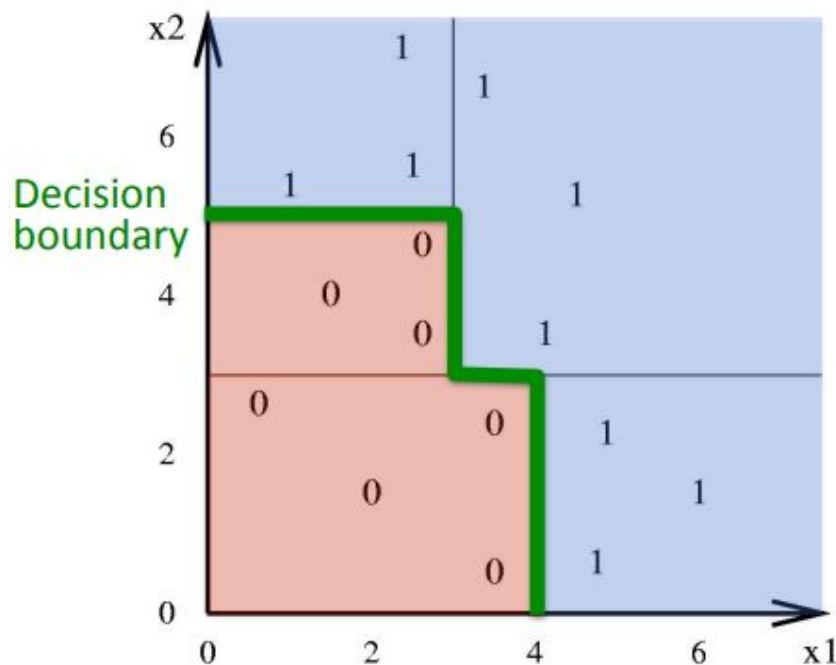
- Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems.
- In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node**.
 - **Decision nodes** are used to make any decision and have multiple branches, whereas
 - **Leaf nodes** are the output of those decisions and do not contain any further branches.
 - **Branches** which stem from the root, represent different options that are available when making a particular decision.

There are two reasons for using the Decision tree

1. Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
2. Output is discrete. There is not estimation or probability.
3. No large data is available. With data (e.g., 200 rows) decision tree can come up with a very impressive good classifier.
4. The logic behind the decision tree can be easily understood because it shows a tree-like structure.

Decision Tree

- Decision trees divide the feature space into axis-parallel (hyper-)rectangles
- Each rectangular region is labeled with one label



Function Approximation



Problem Setting

- Set of possible instances \mathcal{X}
- Set of possible labels \mathcal{Y}
- Unknown target function $f : \mathcal{X} \rightarrow \mathcal{Y}$
- Set of function hypotheses $H = \{h \mid h : \mathcal{X} \rightarrow \mathcal{Y}\}$

Input: Training examples of unknown target function f

$$\{\langle \mathbf{x}_i, y_i \rangle\}_{i=1}^n = \{\langle \mathbf{x}_1, y_1 \rangle, \dots, \langle \mathbf{x}_n, y_n \rangle\}$$

Output: Hypothesis $h \in H$ that best approximates f

Training Data Example: Goal is to Predict When This Player Will Play Tennis?

- Columns denote features X_i
- Rows denote labeled instances $\langle x_i, y_i \rangle$
- Class label denotes whether a tennis game was played

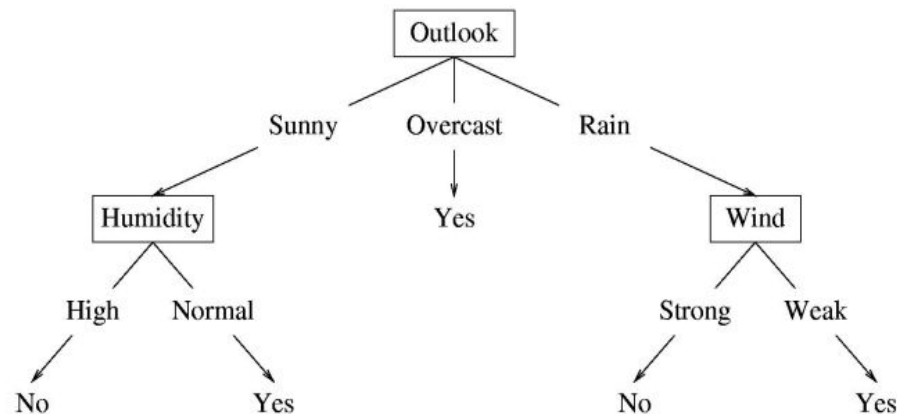
Day	Outlook	Temp	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Weak	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Strong	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

Decision Tree Hypothesis Space

- Each internal node: test one attribute X_i
- Each branch from a node: selects one value for X_i
- Each leaf node: predict Y (or $p(Y \mid \mathbf{x} \in \text{leaf})$)

Suppose the features are **Outlook** (x_1), **Temperature** (x_2), **Humidity** (x_3), and **Wind** (x_4). Then the feature vector $\mathbf{x} = (\text{Sunny}, \text{Hot}, \text{High}, \text{Strong})$ will be classified as **No**. The **Temperature** feature is irrelevant.

A possible decision tree for the data:



CONSTRUCTING DECISION TREES



- The question we need to ask is how, based on those features, we can construct the tree.
- There are a few different decision tree algorithms, but they are almost all variants of the same principle: the algorithms build the tree in a greedy manner starting at the root, choosing the most informative feature at each step.
- We are going to start by focusing on the most common algorithm: [Quinlan's ID3](#), and its extension, known as [C4.5](#), and another classification and regression tree (CART).

Entropy in Information Theory

Entropy: it describes the amount of impurity in a set of features. The higher the entropy more the information content.

- The entropy H of a set of probabilities p_i is $-p_{(+)} \log_2 p_{(+)} - p_{(-)} \log_2 p_{(-)}$
- Where the logarithm is base 2 because we are imagining that we encode everything using binary digits (bits), and we define **0 log 0 = 0**.
- The basic concept is that it tells us how much extra information we would get from knowing the value of that **feature**.
 - If **all of the examples are positive**, then we don't get any extra information from knowing the value of the feature for any particular example, since whatever the value of the feature, the example will be positive. Thus, **the entropy of that feature is 0**.
 - However, if the **feature** separates the **examples into 50% positive and 50% negative**, then the amount of **entropy is at a maximum**, and knowing about that feature is very useful to us.
- For our decision tree, the best feature to pick as the one to classify on now is the one that gives you the most information, i.e., the one with the highest entropy.

Entropy: Example

- Given a training set D , the entropy of D is defined as:

$$Ent(D) = - \sum_{y \in \mathcal{Y}} P(y|D) \log P(y|D).$$

$$Ent(D) = -p_{(+)} \log_2 p_{(+)} - p_{(-)} \log_2 p_{(-)}$$

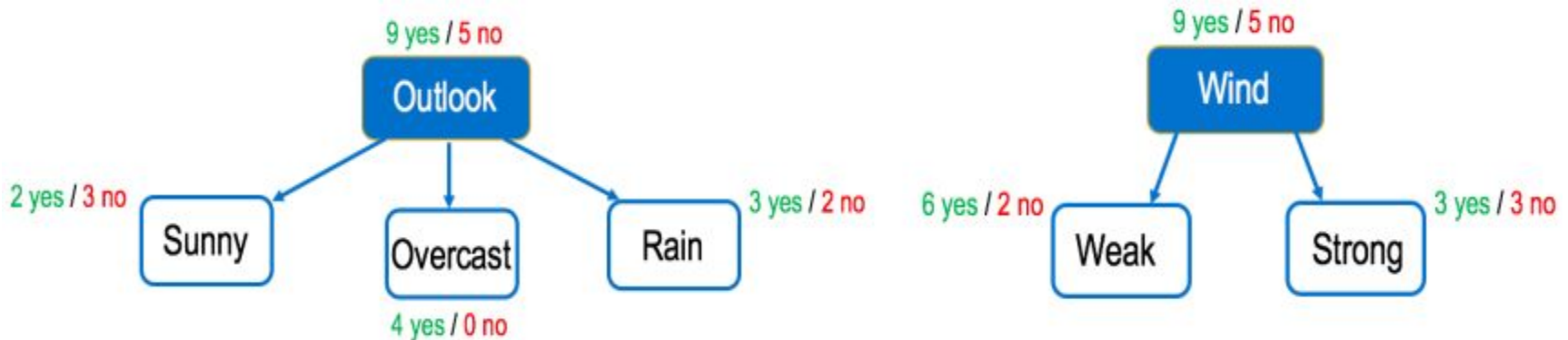
- $p_{(+)}$: (# of yes) / (# of total training examples)
- $p_{(-)}$: (# of no) / (# of total training examples)

Entropy example

- 4 yes / 0 no: $Ent(D) = -1 \log_2 1 - 0 = 0$
- 3 yes / 3 no: $Ent(D) = -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} = 1$

Entropy: Example

Idea: a good attribute splits the examples into subsets that are (ideally) “all positive” or “all negative”



Which split is more informative: **Outlook**? Or **Wind**?

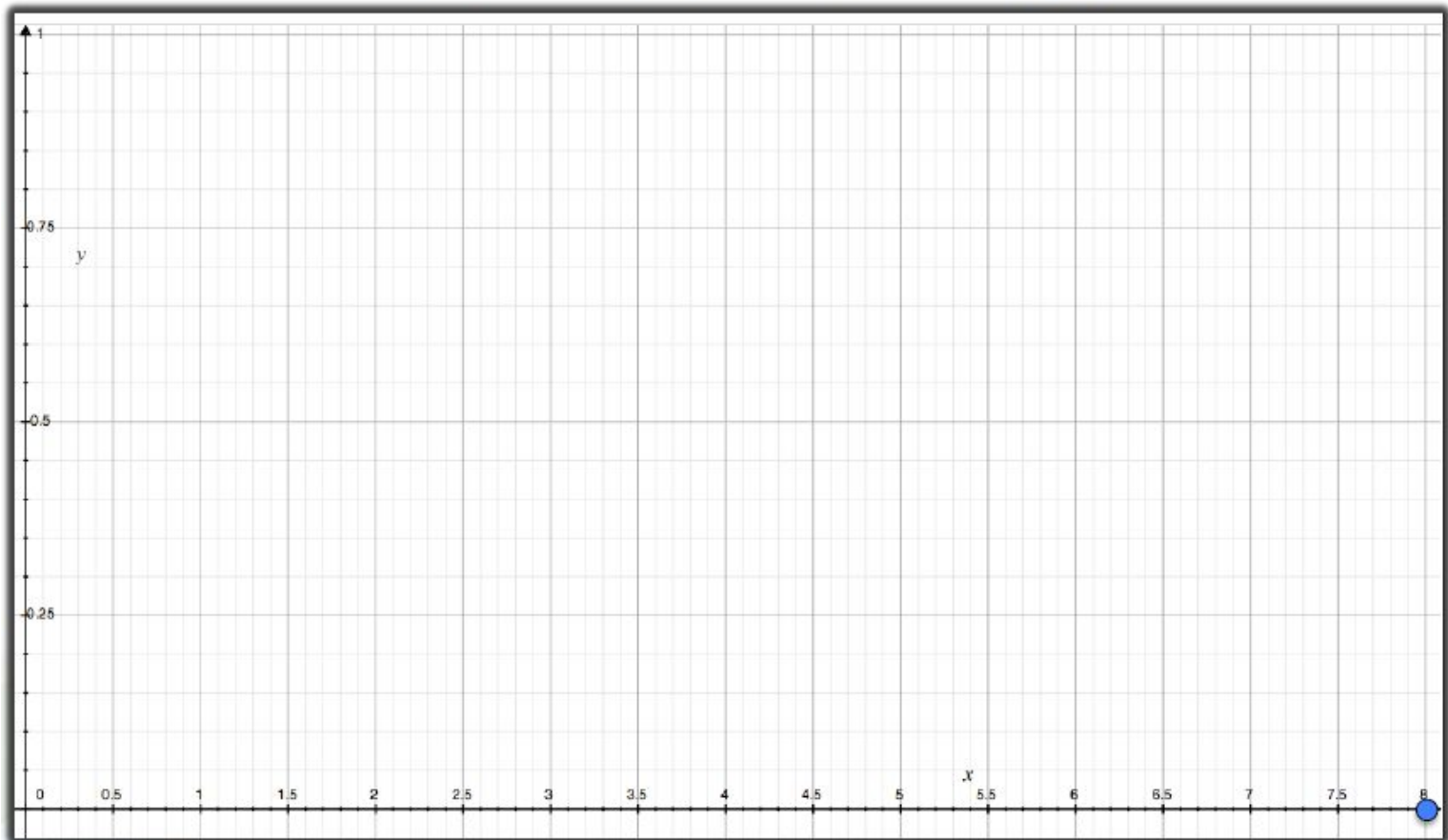
Entropy

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$$

c=red and blue
c=2



$$-\left(\frac{8}{8}\right) \log_2 \left(\frac{8}{8}\right) - \left(\frac{0}{8}\right) \log_2 \left(\frac{0}{8}\right) = 0$$

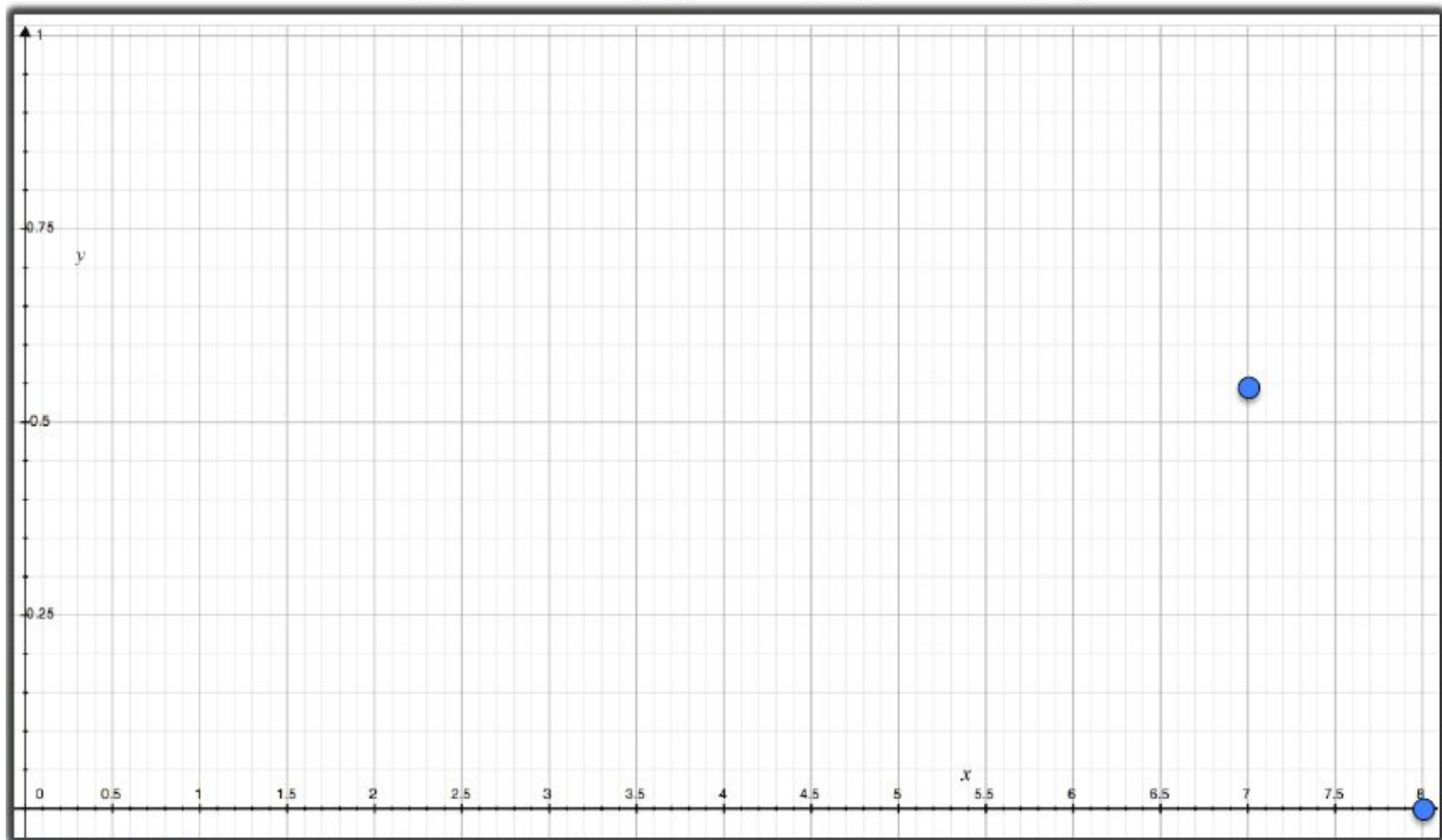


Entropy

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$$



$$-\left(\frac{7}{8}\right) \log_2 \left(\frac{7}{8}\right) - \left(\frac{1}{8}\right) \log_2 \left(\frac{1}{8}\right) = 0.54$$

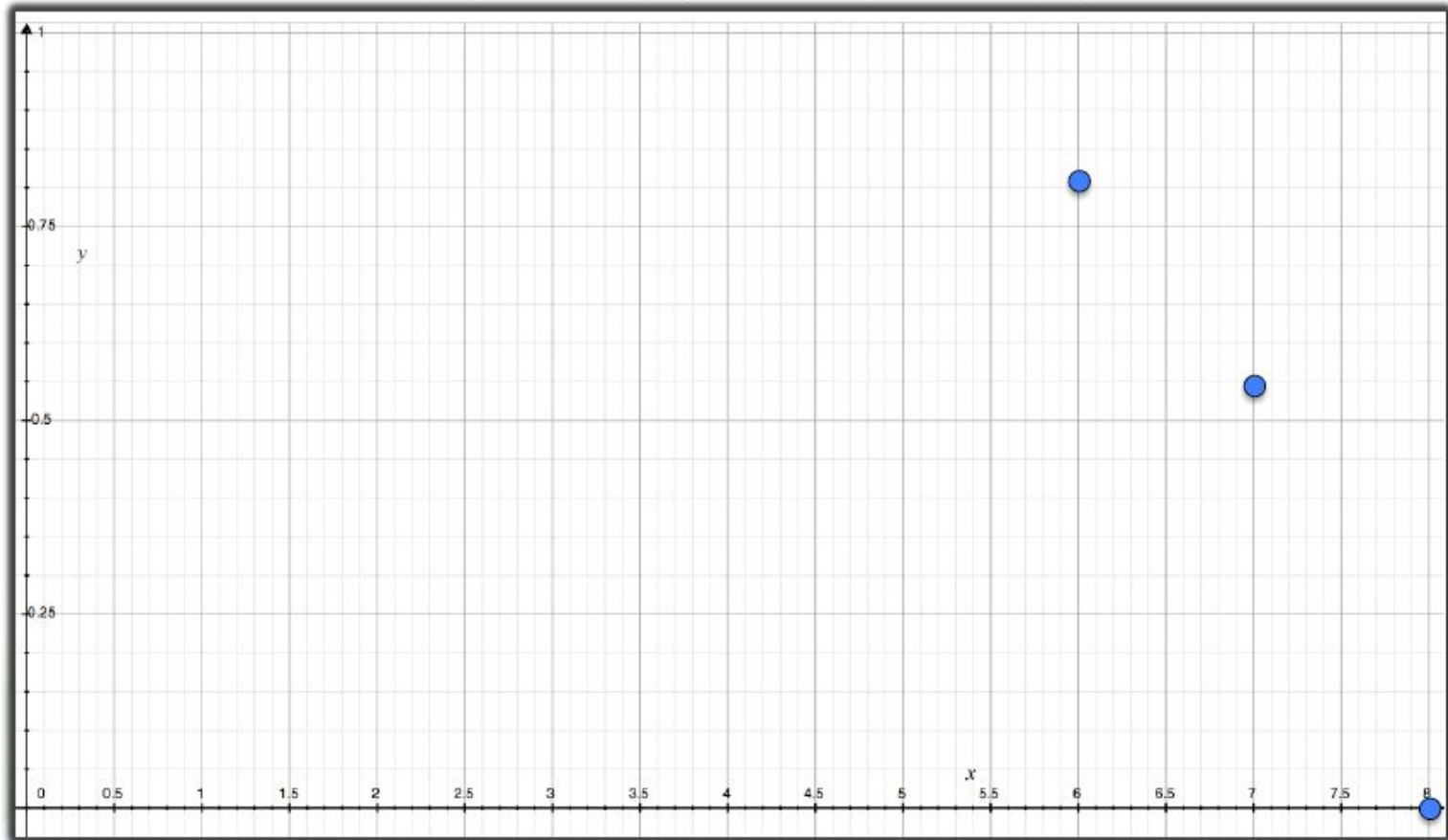


Entropy

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$$



$$-\left(\frac{6}{8}\right) \log_2 \left(\frac{6}{8}\right) - \left(\frac{2}{8}\right) \log_2 \left(\frac{2}{8}\right) = 0.81$$

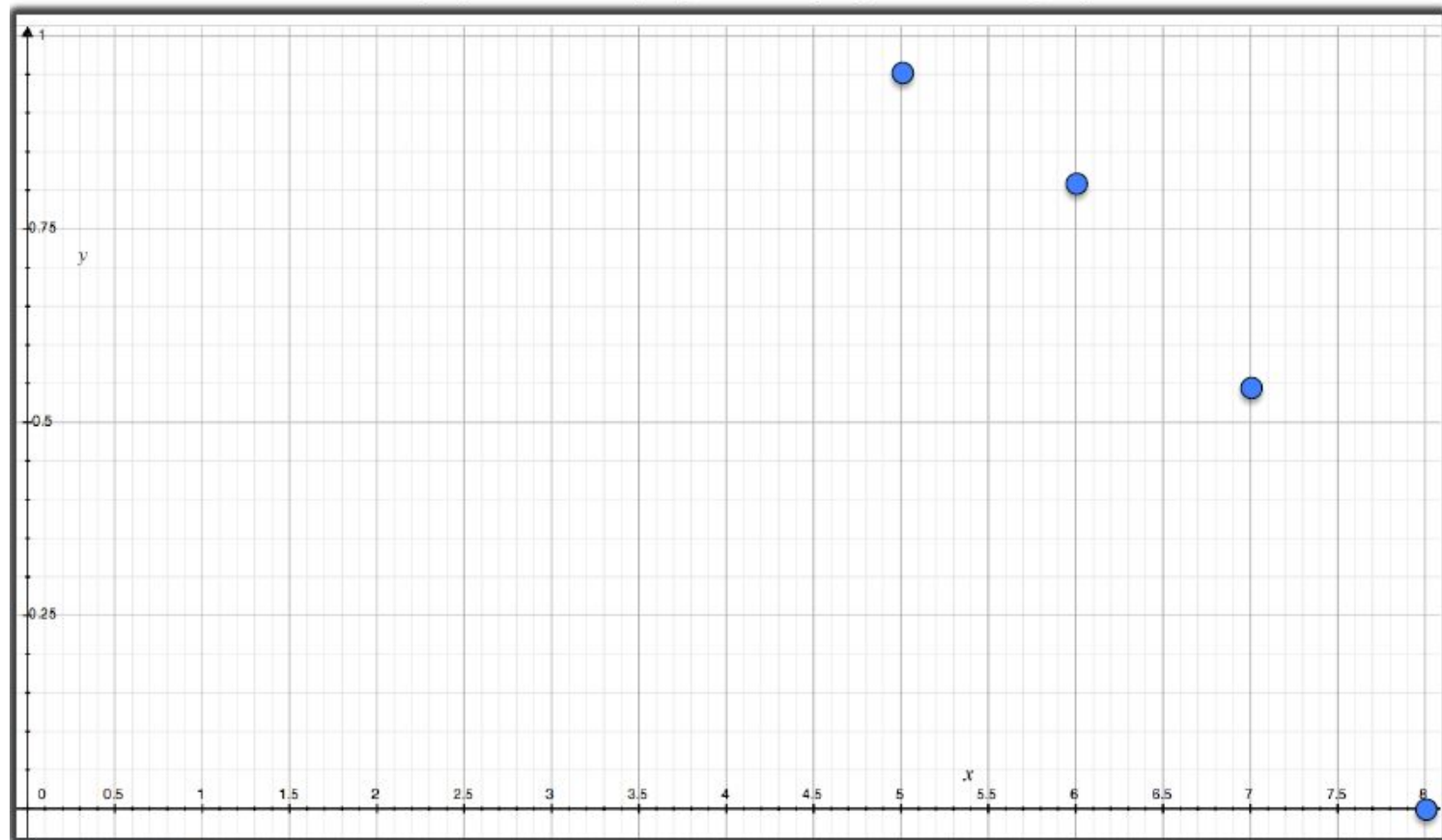


Entropy

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$$



$$-\left(\frac{5}{8}\right) \log_2 \left(\frac{5}{8}\right) - \left(\frac{3}{8}\right) \log_2 \left(\frac{3}{8}\right) = 0.95$$

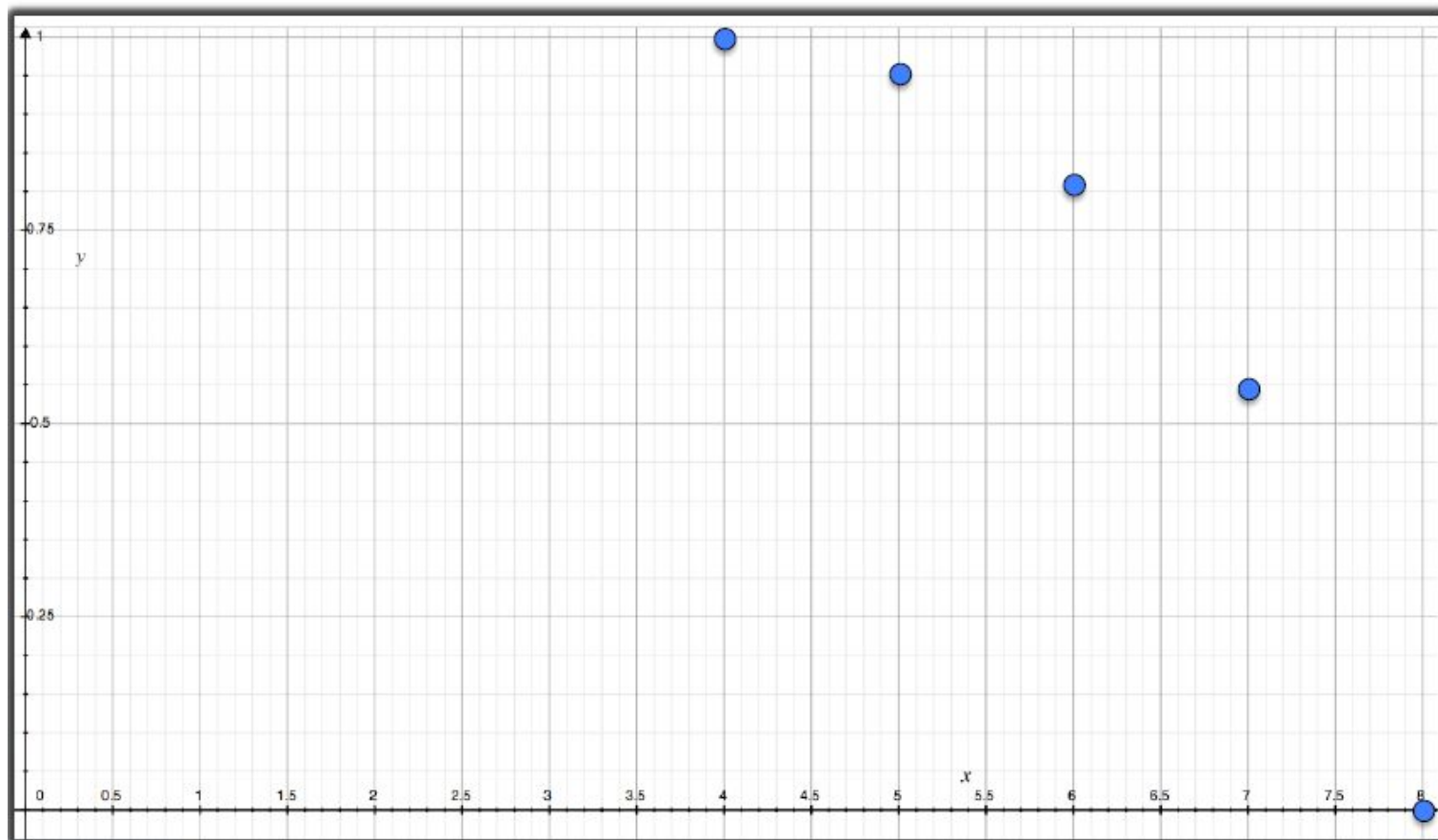


Entropy

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$$



$$-\left(\frac{4}{8}\right) \log_2 \left(\frac{4}{8}\right) - \left(\frac{4}{8}\right) \log_2 \left(\frac{4}{8}\right) = 1$$

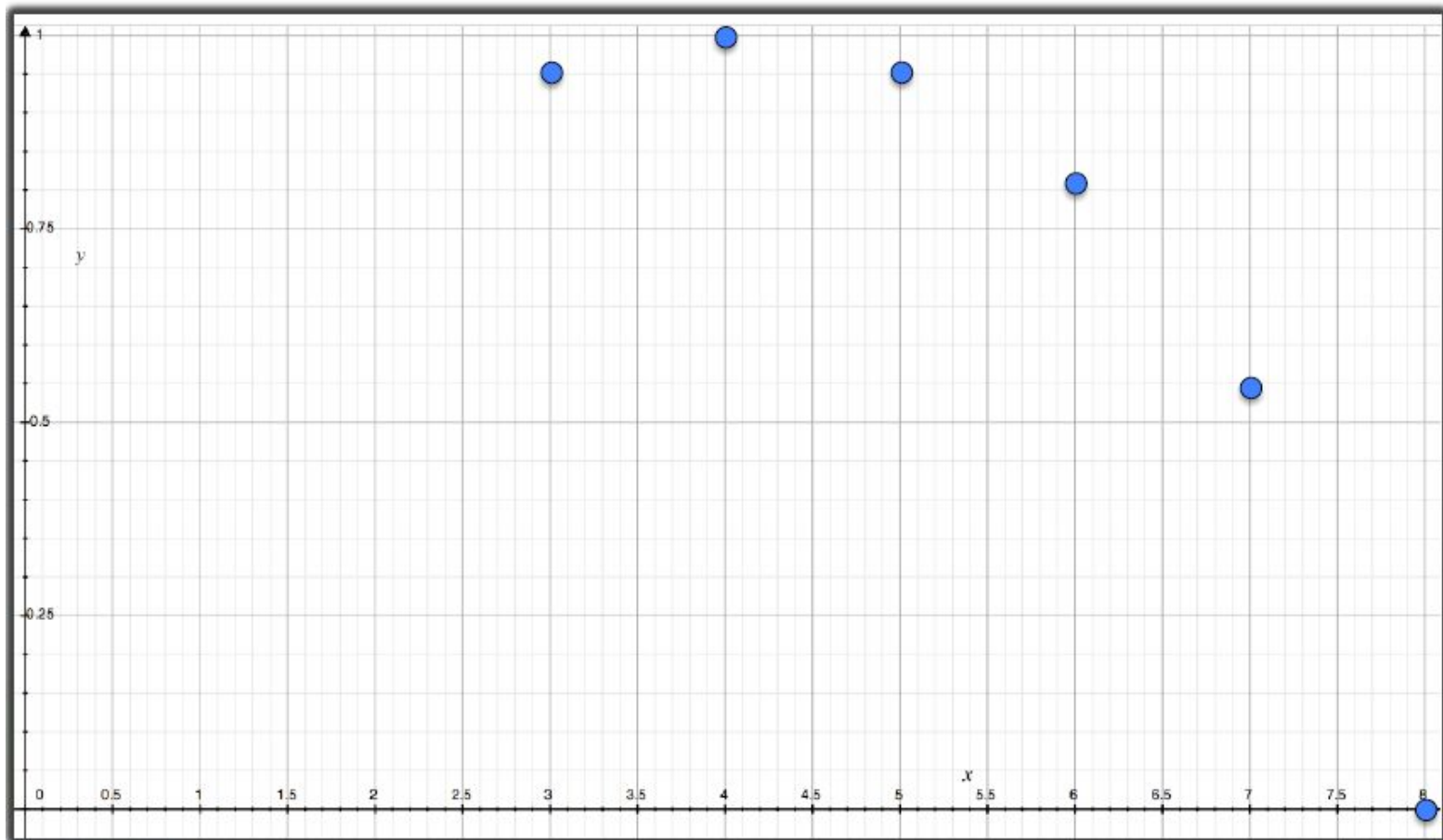


Entropy

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$$



$$-\left(\frac{3}{8}\right) \log_2 \left(\frac{3}{8}\right) - \left(\frac{5}{8}\right) \log_2 \left(\frac{5}{8}\right) = 0.95$$

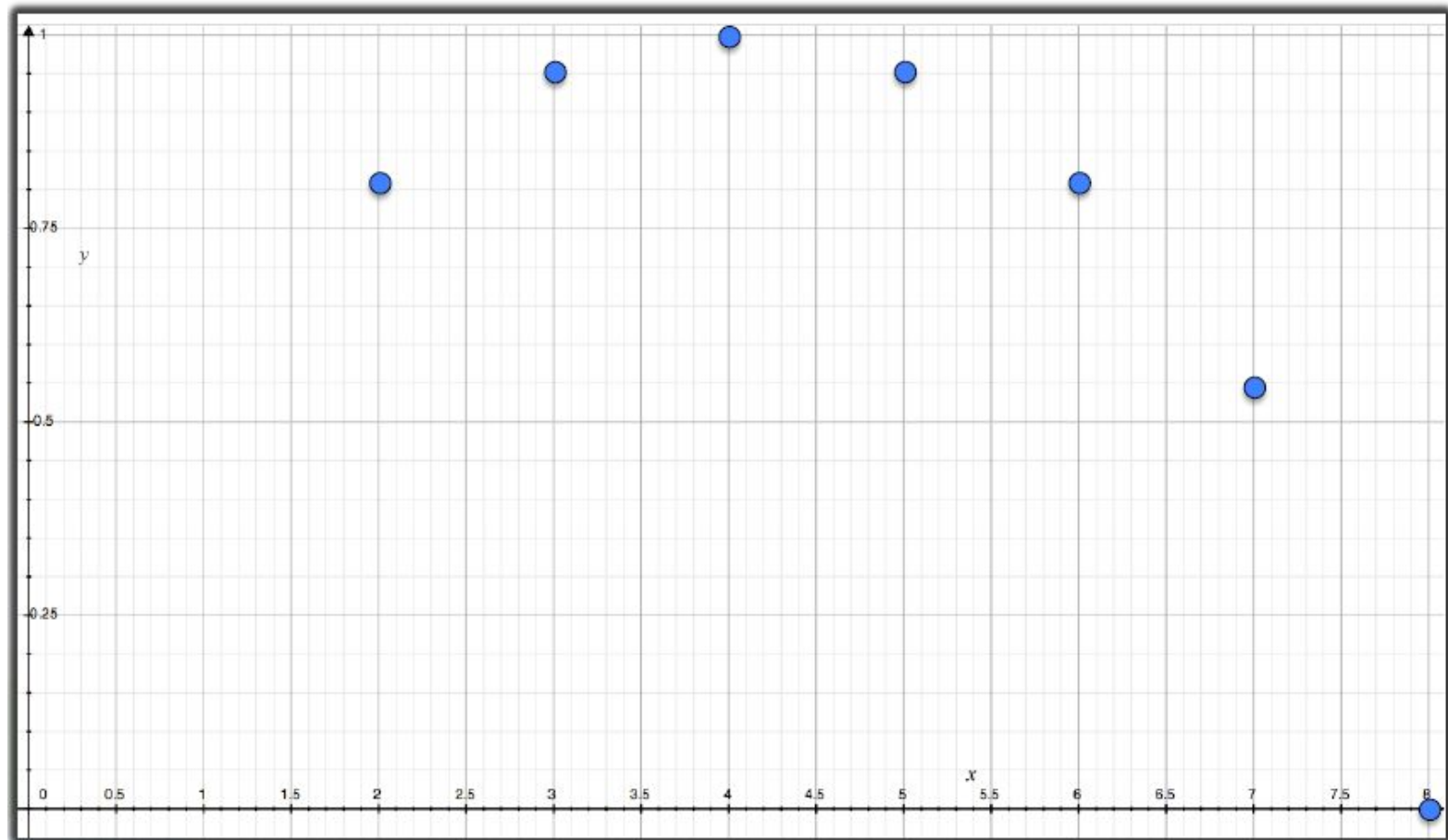


Entropy

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$$



$$-\left(\frac{2}{8}\right) \log_2 \left(\frac{2}{8}\right) - \left(\frac{6}{8}\right) \log_2 \left(\frac{6}{8}\right) = 0.81$$

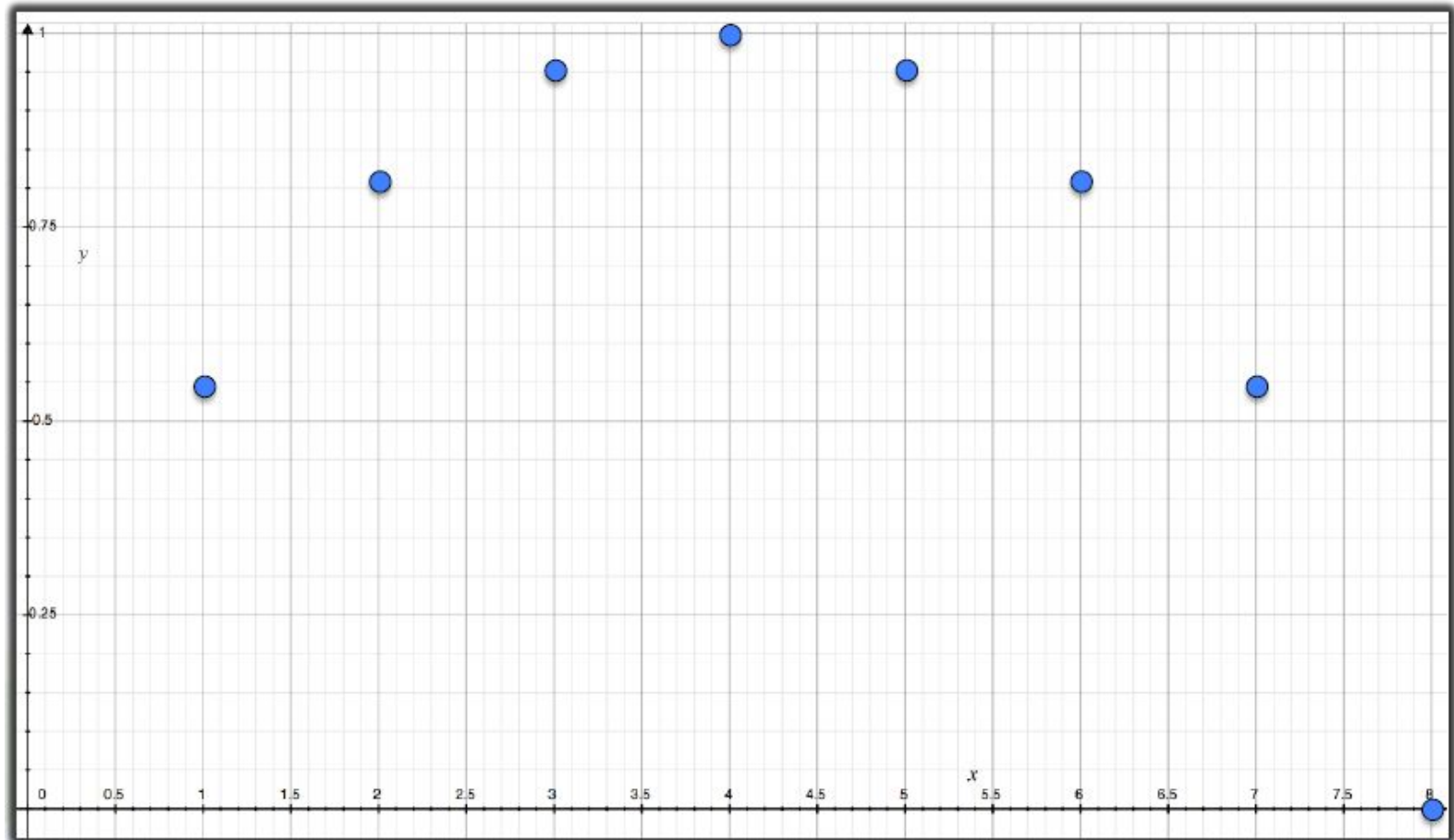


Entropy

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$$



$$-\left(\frac{1}{8}\right) \log_2 \left(\frac{1}{8}\right) - \left(\frac{7}{8}\right) \log_2 \left(\frac{7}{8}\right) = 0.54$$

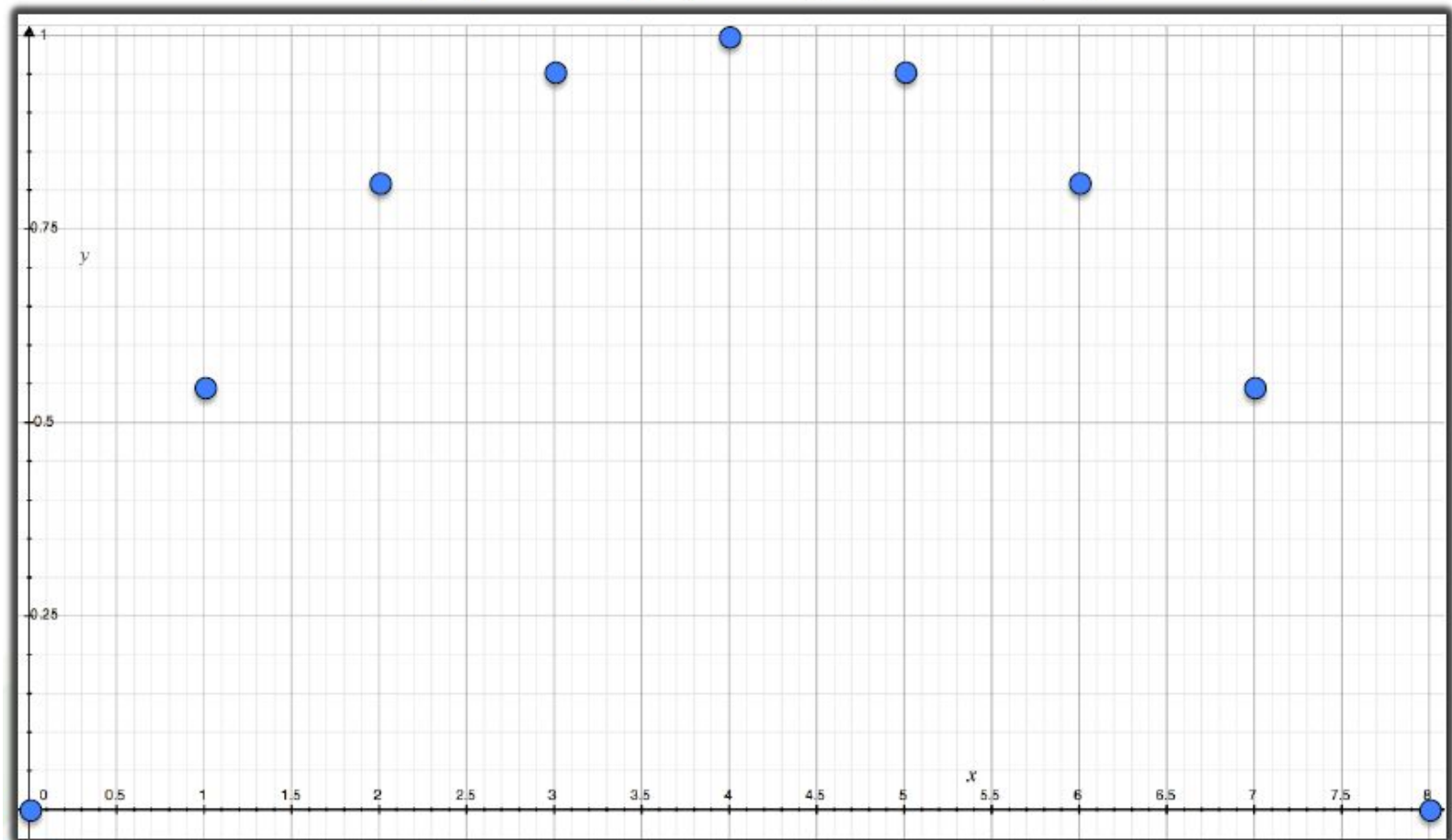


Entropy

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$$

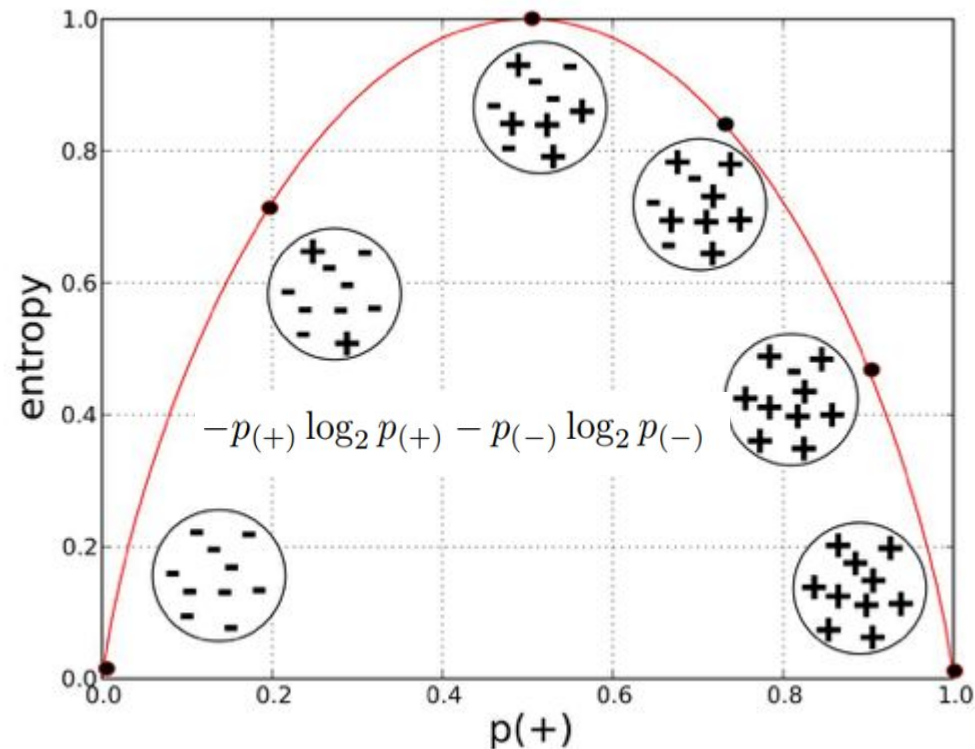


$$-\left(\frac{0}{8}\right) \log_2 \left(\frac{0}{8}\right) - \left(\frac{8}{8}\right) \log_2 \left(\frac{8}{8}\right) = 0$$



Entropy

Entropy: it describes the amount of impurity in a set of features. The higher the entropy more the information content.



Information gain

- Now that we have a suitable measure for choosing which feature to choose next, entropy, we just have to work out how to apply it.
- The important idea is to work out **how much the entropy of the whole training set would decrease** if we choose each particular feature for the next classification step.
- This is known as the **information gain**, and it is defined as the entropy of the whole set minus the entropy when a particular feature is chosen.
- The information gain ($\text{Gain}(S, A)$) of an attribute A relative to a collection of data set S , is defined as

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v) \quad (1.5)$$

Where, $\text{Values}(A)$ is the all possible values for attribute A , and S_v is the subset of S for which attribute A has value v .

Information gain

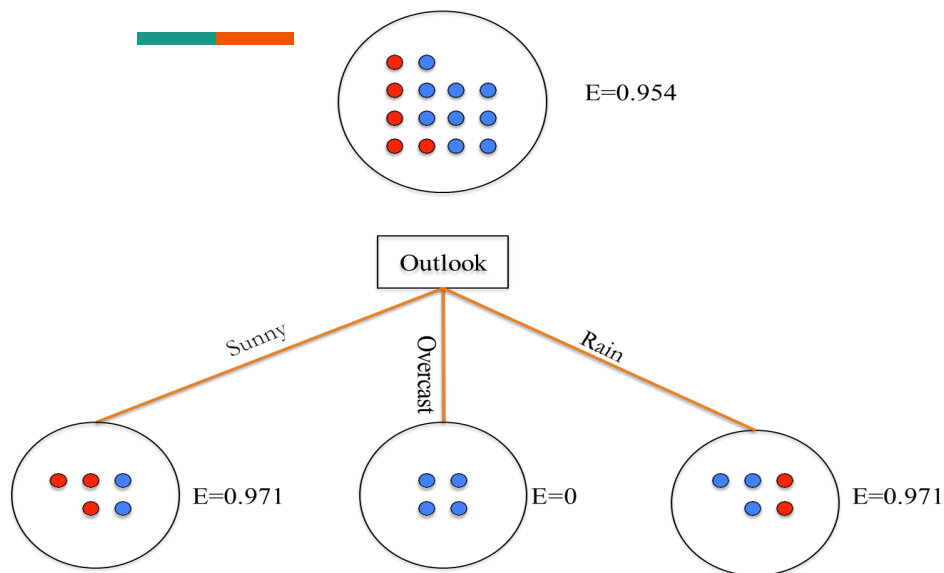
- Given **Entropy** is the measure of impurity in a collection of a dataset, now we can measure the effectiveness of an attribute in classifying the training set.
- The measure we will use called **information gain**, is simply the expected reduction in entropy caused by partitioning the data set according to this attribute.
- The **information gain** ($\text{Gain}(S, A)$) of an attribute A relative to a collection of data set S , is defined as

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v) \quad (1.5)$$

Where, $\text{Values}(A)$ is the all possible values for attribute A , and S_v is the subset of S for which attribute A has value v .

- The important idea is to work out **how much the entropy of the whole training set would decrease** if we choose each particular feature for the next classification step.

Entropy(Outlook)



$$-(3 \div 5) \times -0.737 - (2 \div 5) \times -1.322 = 0.971 \text{ (sunny)}$$

$$-(4 \div 4) \times -0 - (0 \div 5) \times \log(0) = 0 \text{ (overcast)}$$

$$-(3 \div 5) \times -0.737 - (2 \div 5) \times -1.322 = 0.971 \text{ (Rain)}$$

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$$

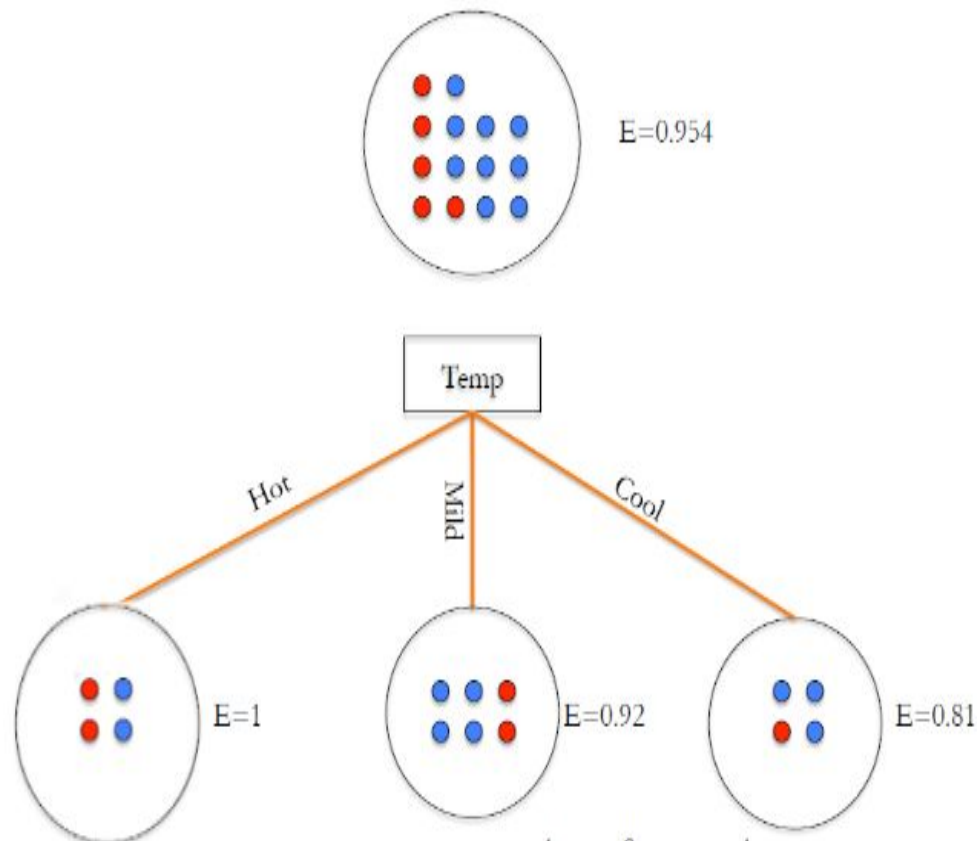
Day	Outlook	Temp	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Weak	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Strong	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

Entropy: it describes the amount of impurity in a set of features. The higher the entropy more the information content.

Entropy(Temp)

$$-(5 \div 14) \times (-1.4854) - (9 \div 14) \times -0.6374 =$$

0.94025714285



$$-2 \div 4 \times (-1) - (2 \div 4) \times (-1) =$$

1

$$-4 \div 6 \times -0.585 - 2 \div 6 \times -1.585 =$$

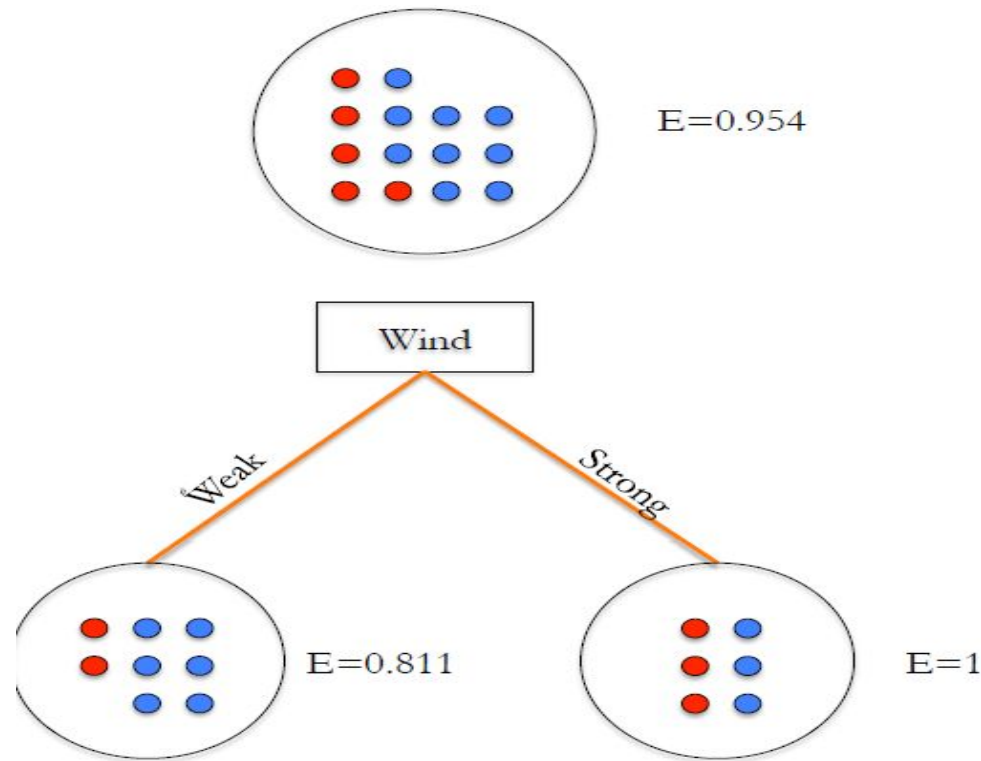
0.91833333333

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$$

Day	Outlook	Temp	Humidity	Wind	Play 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	Weak	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Strong	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

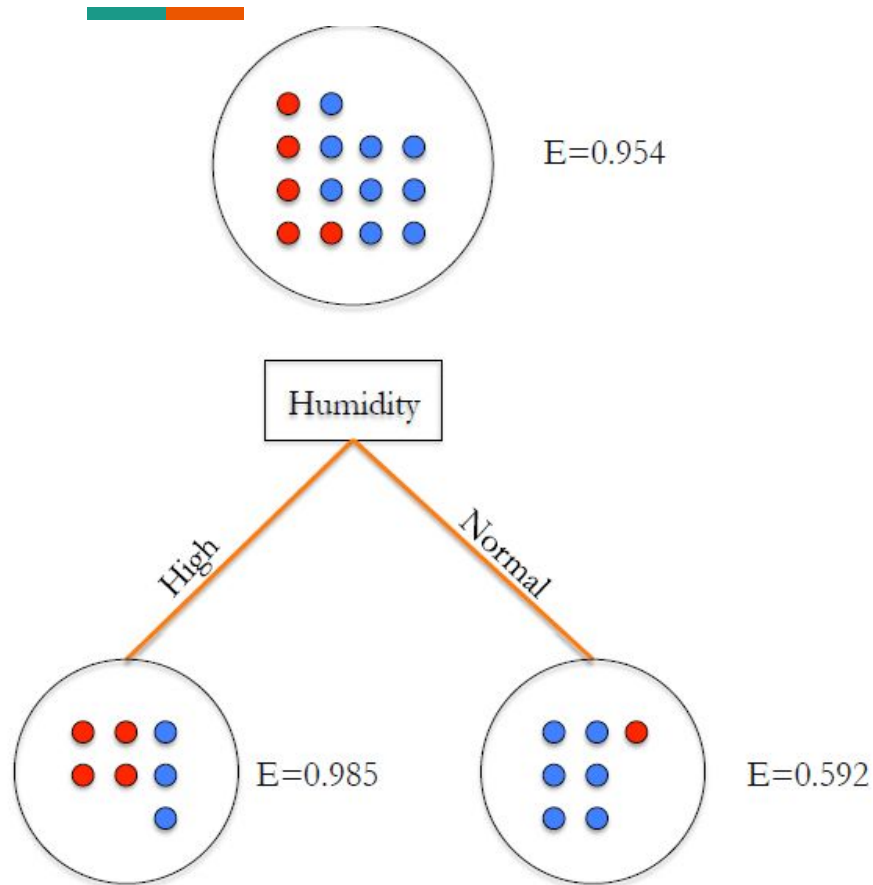
$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$$

Entropy (wind)



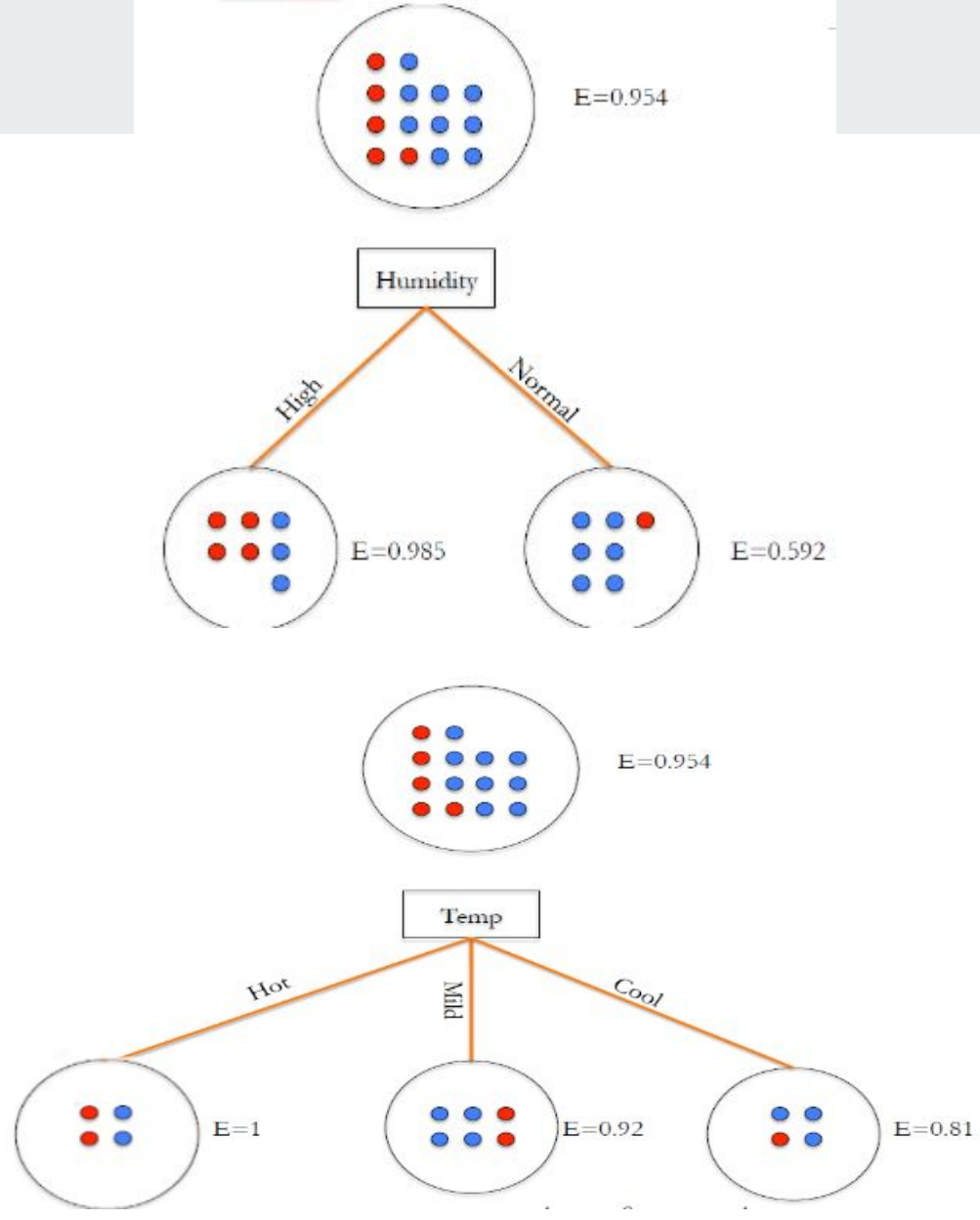
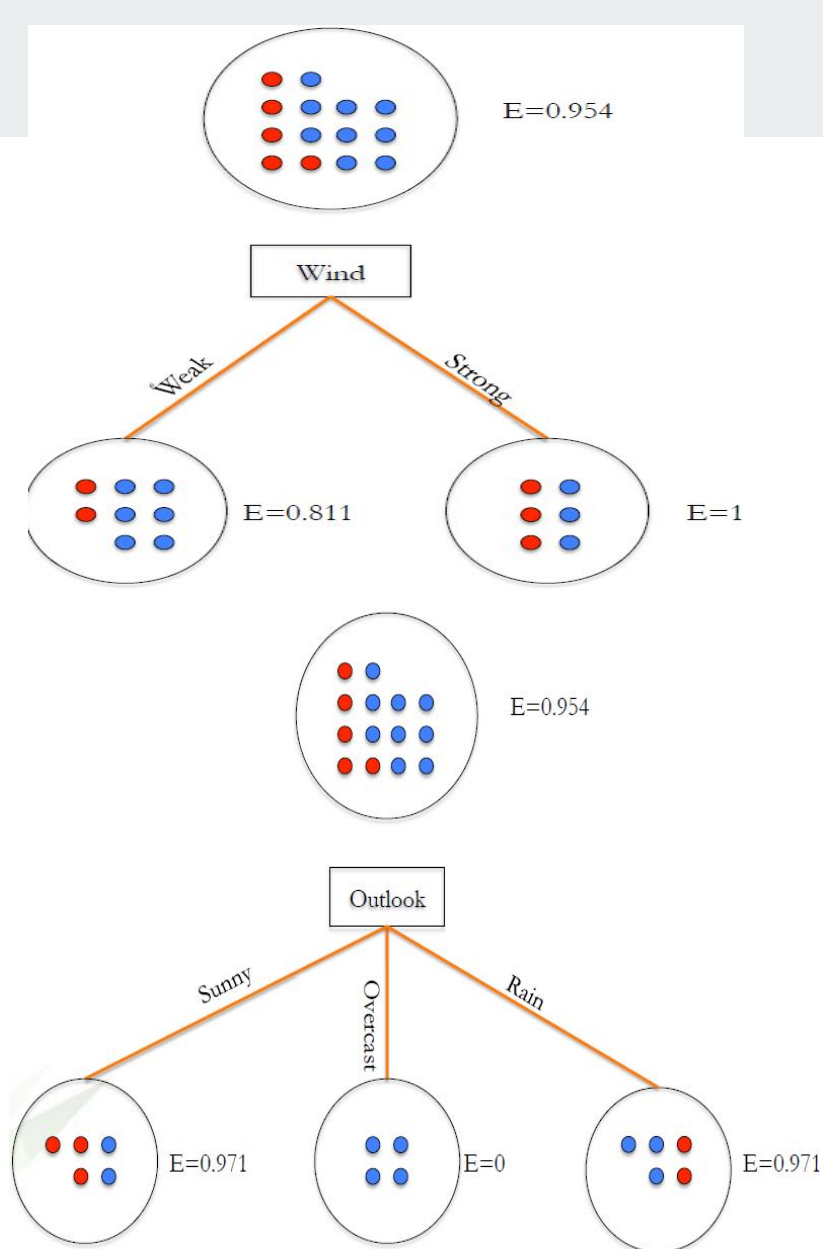
Day	Outlook	Temp	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Weak	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Strong	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

Entropy (Humidity)



$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$$

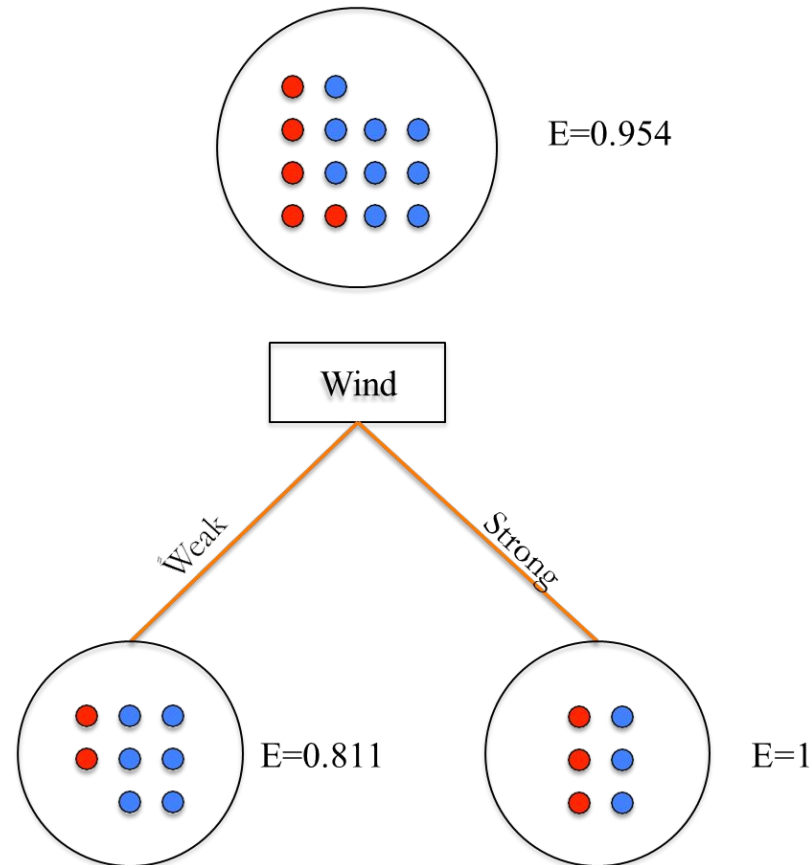
Day	Outlook	Temp	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Weak	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Strong	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



Which split is more informative: **Wind?**, **Outlook?**, **Humidity?** Or **Temp?**

Information gain

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$



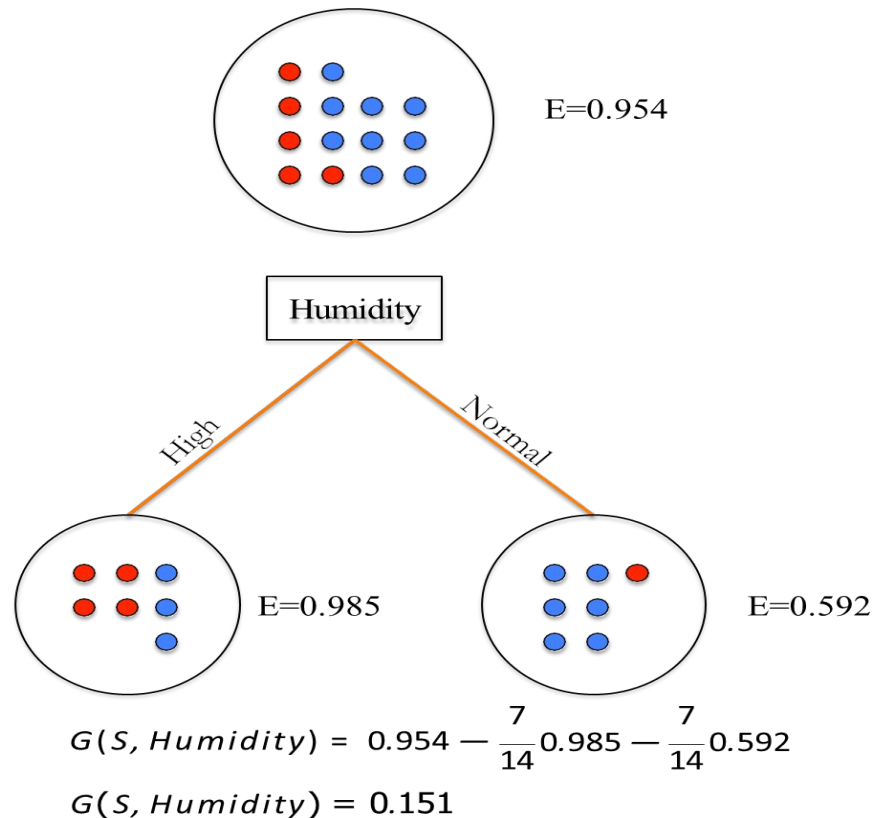
$$G(S, Wind) = 0.954 - \frac{8}{14} 0.811 - \frac{6}{14} 1$$

$$G(S, Wind) = 0.048$$

Information gain

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$G(S, Wind) = 0.048$$



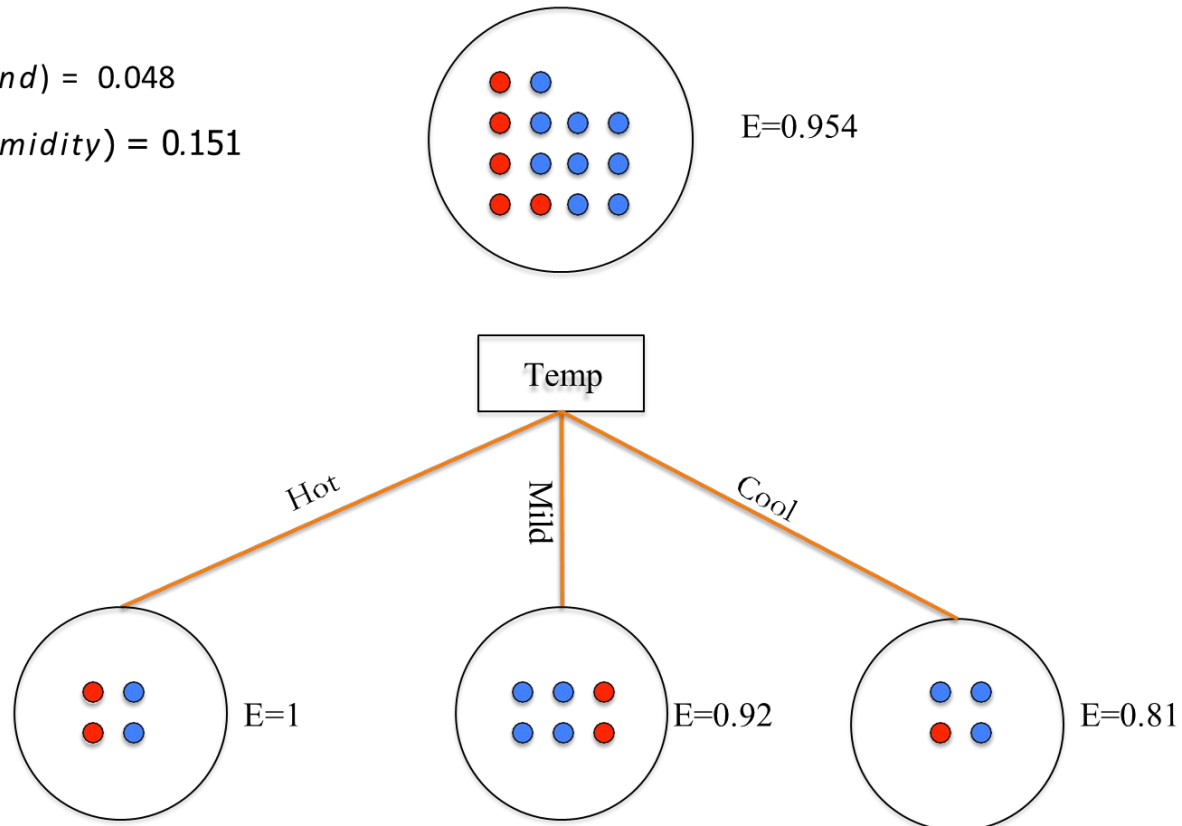
- The important idea is to work out **how much the entropy of the whole training set would decrease** if we choose each particular feature for the next classification step.

Information gain

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$G(S, Wind) = 0.048$$

$$G(S, Humidity) = 0.151$$



$$G(S, Temp) = 0.954 - \frac{4}{14}1 - \frac{6}{14}0.92 - \frac{4}{14}0.81$$

$$G(S, Temp) = 0.042$$

Information gain

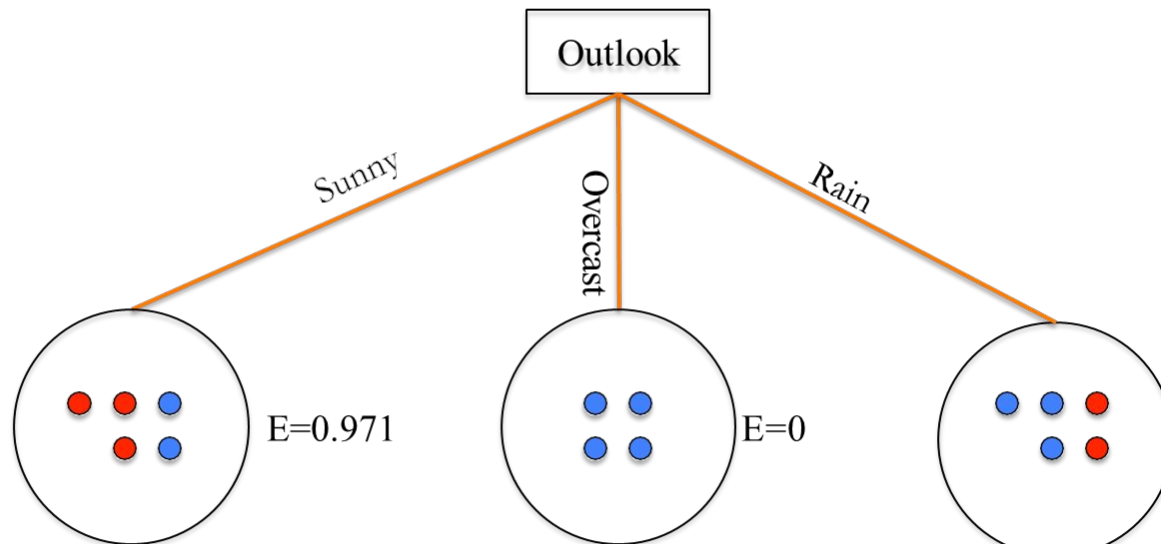
$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

$G(S, Wind) = 0.048$

$G(S, Humidity) = 0.151$

$G(S, Temp) = 0.042$

$E=0.954$



$$G(S, Outlook) = 0.954 - \frac{5}{14}0.971 - \frac{4}{14}0 - \frac{5}{14}0.971$$

$G(S, Outlook) = 0.247$

Information gain

$$G(S, Q) = E(S) - \sum_{i=1}^k p_i E(S, Q_i)$$

$$G(S, Wind) = 0.048$$

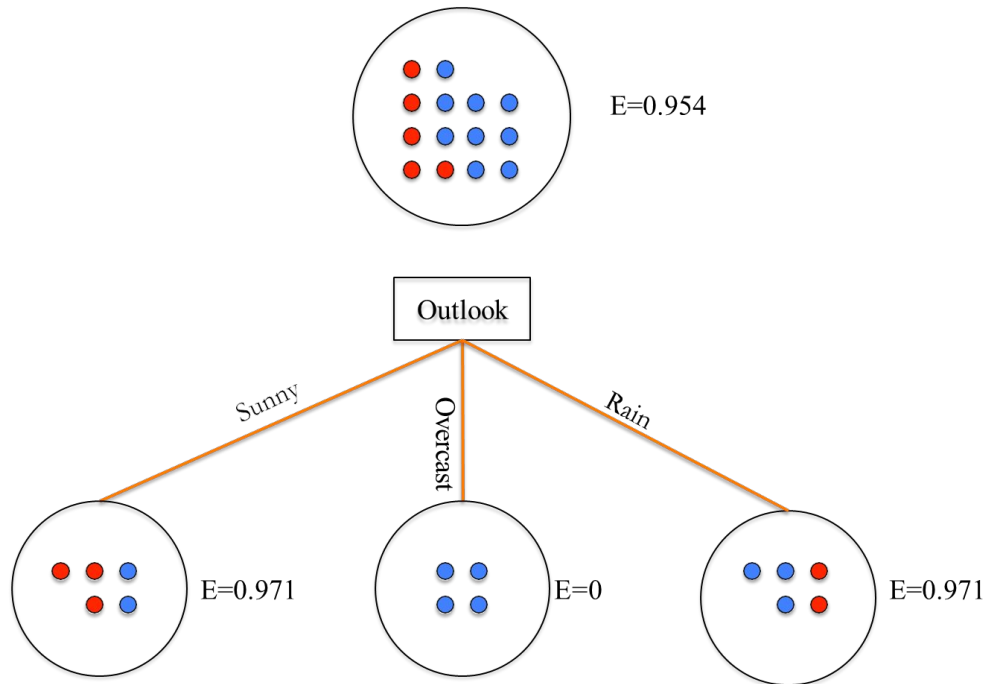
$$G(S, Humidity) = 0.151$$

$$G(S, Temp) = 0.042$$

$$G(S, Outlook) = 0.247 \quad \leftarrow$$

A larger **information gain** suggests a lower entropy group of samples.

Which attribute should be tested in the leaf nodes?

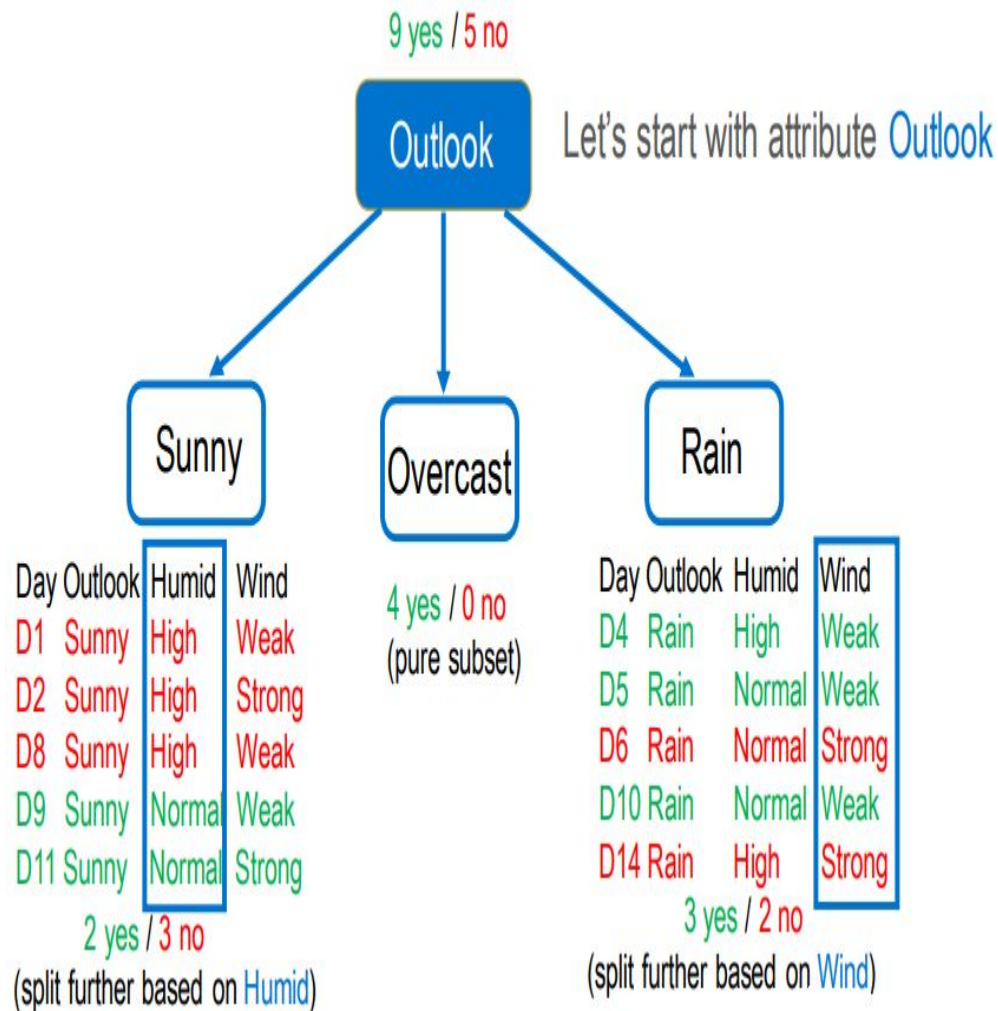


$$Ent(D) = - \sum_{u \in \mathcal{V}} P(y|D) \log P(y|D).$$

$$G(S, Q) = E(S) - \sum_{i=1}^k p_i E(S, Q_i)$$

Day	Outlook	Temp	Humidity	Wind	Play 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	Weak	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Strong	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

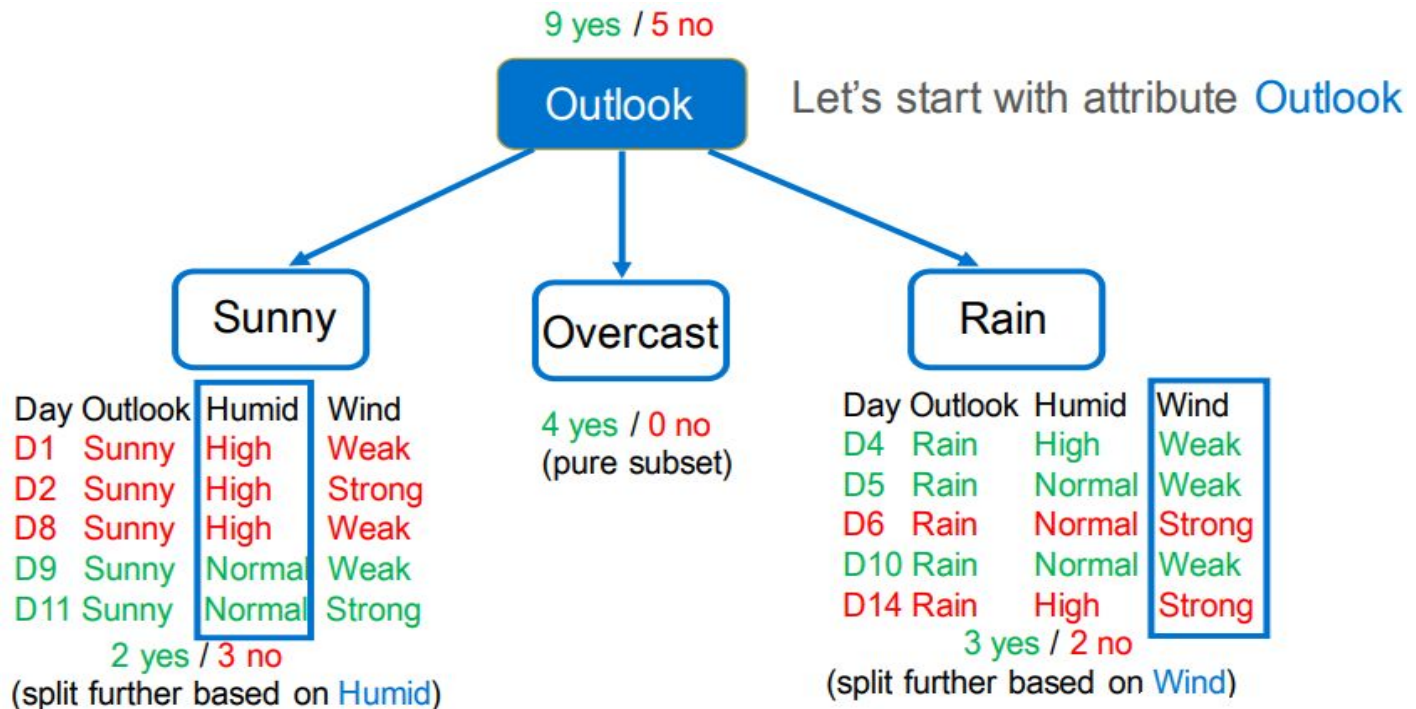
Which attribute should be tested in the leaf nodes?



Day	Outlook	Temp	Humidity	Wind	Play 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	Weak	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Strong	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

Information gain

$$G(S, Q) = E(S) - \sum_{i=1}^k p_i E(S, Q_i)$$



Which attribute should be tested here?

$$S_{\text{sunny}} = \{D1, D2, D8, D9, D11\}$$

$$\text{Gain}(S_{\text{sunny}}, \text{Humidity}) = .970 - (3/5) 0.0 - (2/5) 0.0 = .970$$

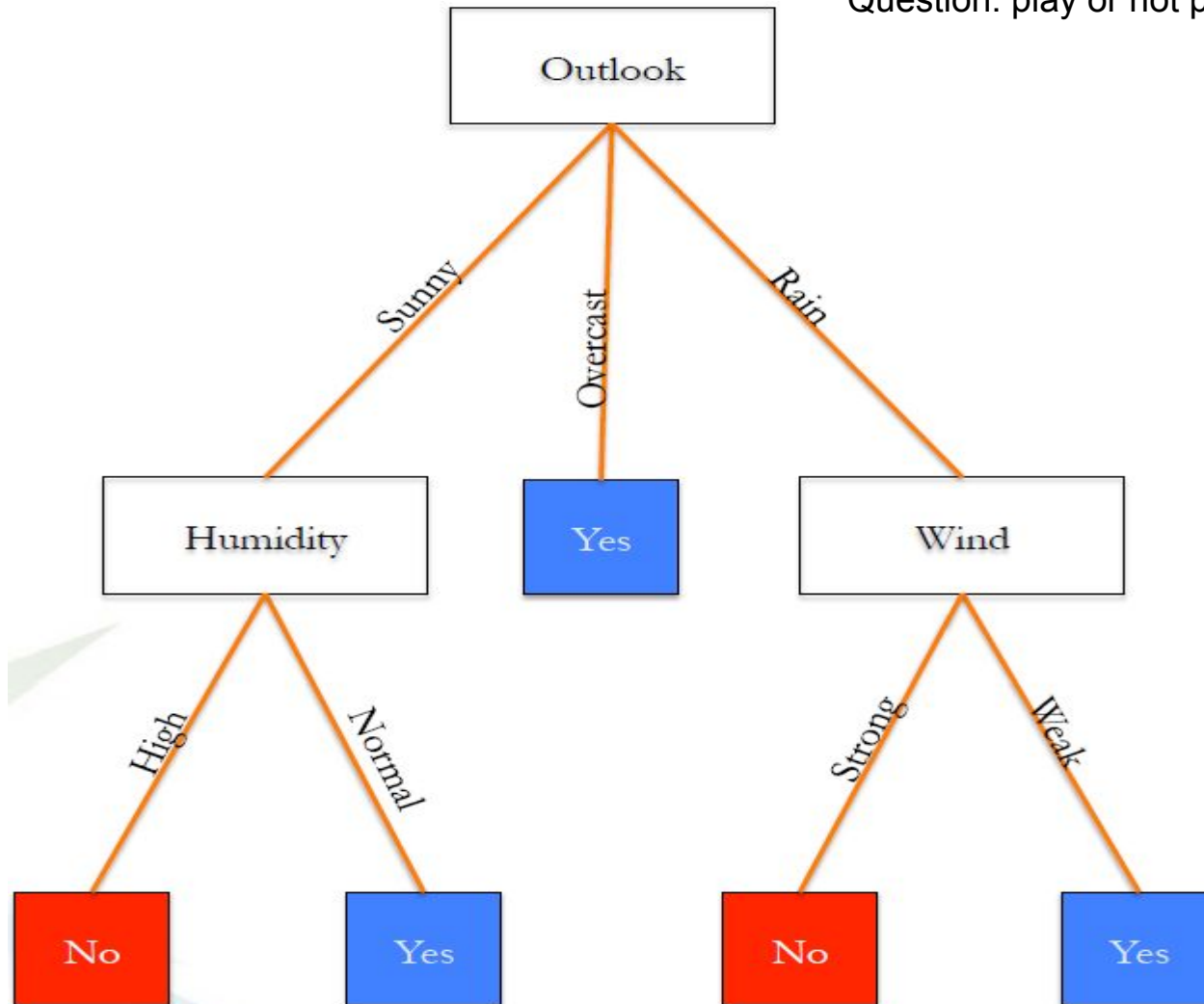
$$\text{Gain}(S_{\text{sunny}}, \text{Temperature}) = .970 - (2/5) 0.0 - (2/5) 1.0 - (1/5) 0.0 = .570$$

$$\text{Gain}(S_{\text{sunny}}, \text{Wind}) = .970 - (2/5) 1.0 - (3/5) .918 = .019$$


Decision Tree (Example)



Test data attributes: Rain High Weak
Question: play or not play? (Guess?)




ID3 Algorithm Steps

- 
- one of the many Algorithms used to build Decision Trees (ID3, CART, CHAID, etc.)
 - ID3 stands for Iterative Dichotomiser 3 and is named such because the algorithm iteratively (repeatedly) dichotomizes(divides) features into two or more groups at each step.
 - the ID3 algorithm selects the best feature at each step while building a Decision Tree.


ID3 Steps

1. Calculate the Information Gain of each feature.
2. Considering that all rows don't belong to the same class, split the dataset S into subsets using the feature for which the Information Gain is maximum.
3. Make a decision tree node using the feature with the maximum Information gain.
4. If all rows belong to the same class, make the current node as a leaf node with the class as its label.
5. Repeat for the remaining features until we run out of all features, or the decision tree has all leaf nodes.

Decision Tree Pruning Why is it Important?

- 
- Pruning is a technique that removes the parts of the Decision Tree which prevent it from growing to its full depth.
 - The parts that it removes from the tree are the parts that do not provide the power to classify instances.
 - A Decision tree that is trained to its full depth will highly likely lead to overfitting the training data - therefore Pruning is important.
 - In simpler terms, the aim of Decision Tree Pruning is to construct an algorithm that will perform worse on training data but will generalize better on test data.
 - Tuning the hyperparameters of your Decision Tree model can do your model a lot of justice and save you a lot of time and money.

How do you Prune a Decision Tree?



There are two types of pruning: Pre-pruning and Post-pruning. I will go through both of them and how they work.

1. **Pre-pruning.** The pre-pruning technique of Decision Trees is tuning the hyperparameters prior to the training.
2. **Post-pruning:** The post-pruning does the opposite of pre-pruning and allows the Decision Tree model to grow to its full depth. Once the model grows to its full depth, tree branches are removed to prevent the model from overfitting.

References



<https://www.cise.ufl.edu/~ddd/cap6635/Fall-97/Short-papers/2.htm>

[Machine Learning: An Algorithmic Perspective, Second Edition \(Chapman & Hall/CRC Machine Learning & Pattern Recognition\) 2nd Edition](#)