CORK INSTITUTE OF TECHNOLOGY
INSTITIÚID TEICNEOLAÍOCHTA CHORCAÍ

Machine Learning

Machine Learning

Lecture: Decision Trees

Ted Scully

Decision Trees

- A decision tree is an easily interpretable machine learning model (we can represent its **decision process** graphically in tree form).

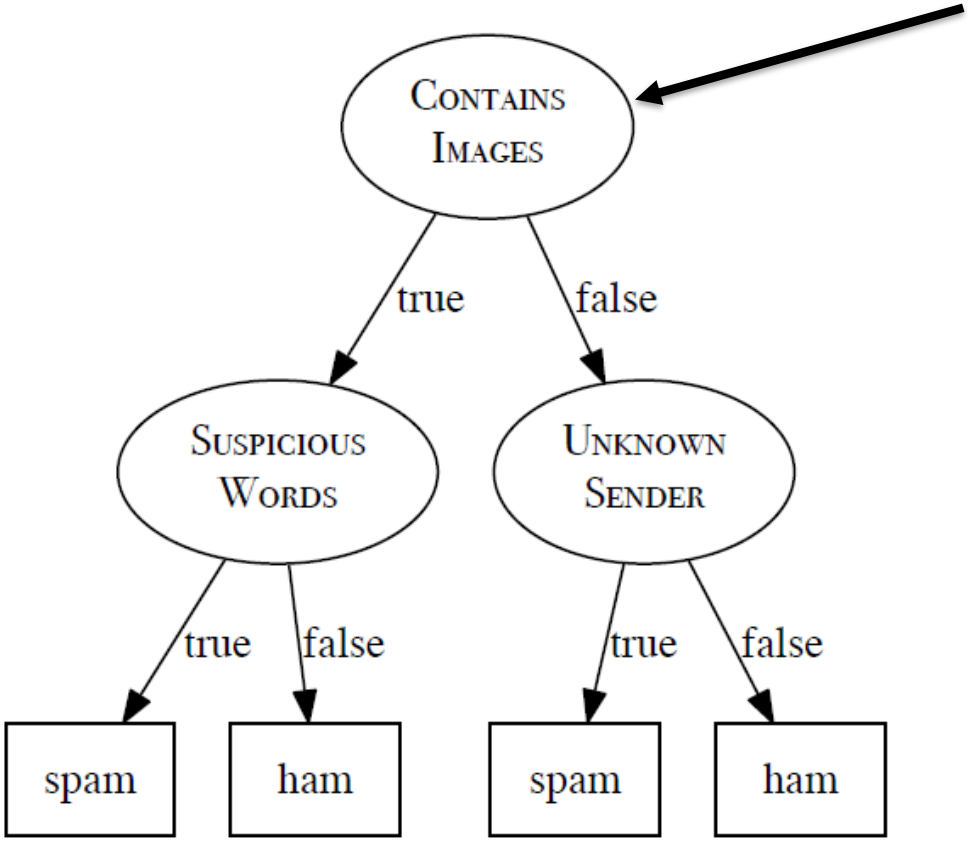
- A decision tree consists of:
  - a root node (or starting node),
  - interior nodes
  - and leaf nodes (or terminating nodes).

- The **root** and **interior** nodes in the decision tree represent the features of your dataset and the leaf nodes represent the predicted target class/regression value.

- Each of the **non-leaf nodes (root and interior)** in the tree specifies a test to be carried out on one of the query's features.

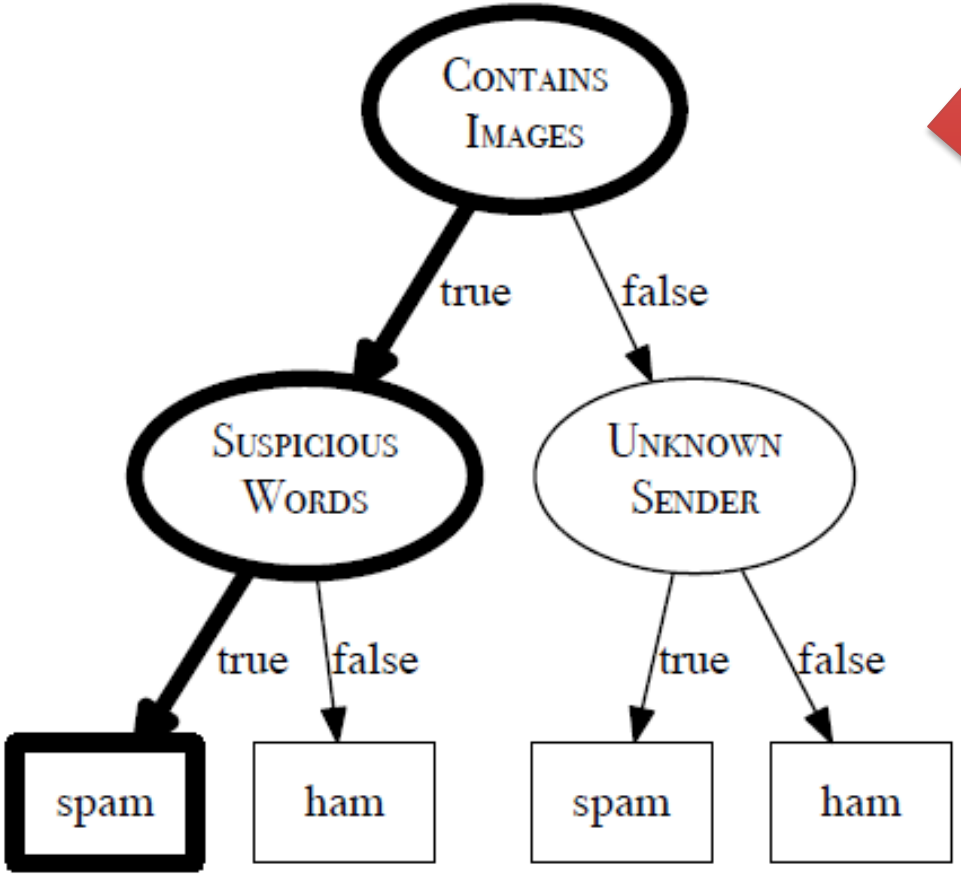
- Each of the **leaf nodes** specifies a predicted classification for the query.

Suspicious Words	Unknown Sender	Contains Images	Class
true	false	true	spam
true	true	false	spam
true	true	false	spam
false	true	true	ham
false	false	false	ham
false	false	false	ham



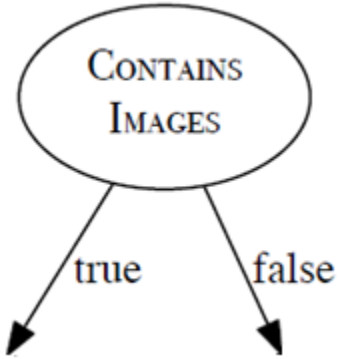
Remember each non-leaf node in a tree is a test on the data. The first test on the data here is on the feature “Contains Images”.

Suspicious Words	Unknown Sender	Contains Images	Class
true	false	true	spam
true	true	false	spam
true	true	false	spam
false	true	true	ham
false	false	false	ham
false	false	false	ham



The process of using a decision tree to make a prediction for a query instance starts by testing the value of the descriptive feature at the root node of the tree. The result then determines which of the two routes we should descend. This process is repeated until we reach a leaf (a prediction).

Suspicious Words	Unknown Sender	Contains Images	Class
true	false	true	spam
true	true	false	spam
true	true	false	spam
false	true	true	ham
false	false	false	ham
false	false	false	ham



The process of **building a decision tree** is highly iterative.

We pick a feature to act as our node (more on this later), this serves to partition the training data.

We then go on to create separate nodes for each partition of the data.

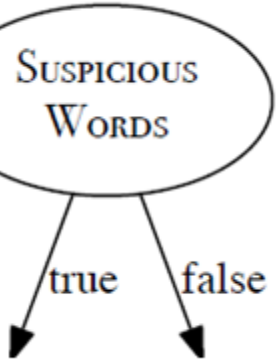
Suspicious Words	Unknown Sender	Contains Images	Class
true	false	true	spam
false	true	true	ham

Suspicious Words	Unknown Sender	Contains Images	Class
true	true	false	spam
true	true	false	spam
false	false	false	ham
false	false	false	ham

Suspicious Words	Unknown Sender	Contains Images	Class
true	false	true	spam
true	true	false	spam
true	true	false	spam
false	true	true	ham
false	false	false	ham
false	false	false	ham

In this example we will illustrate the case where we use suspicious words as the root node.

Notice how using Suspicious Words perfectly classifies the Class label.

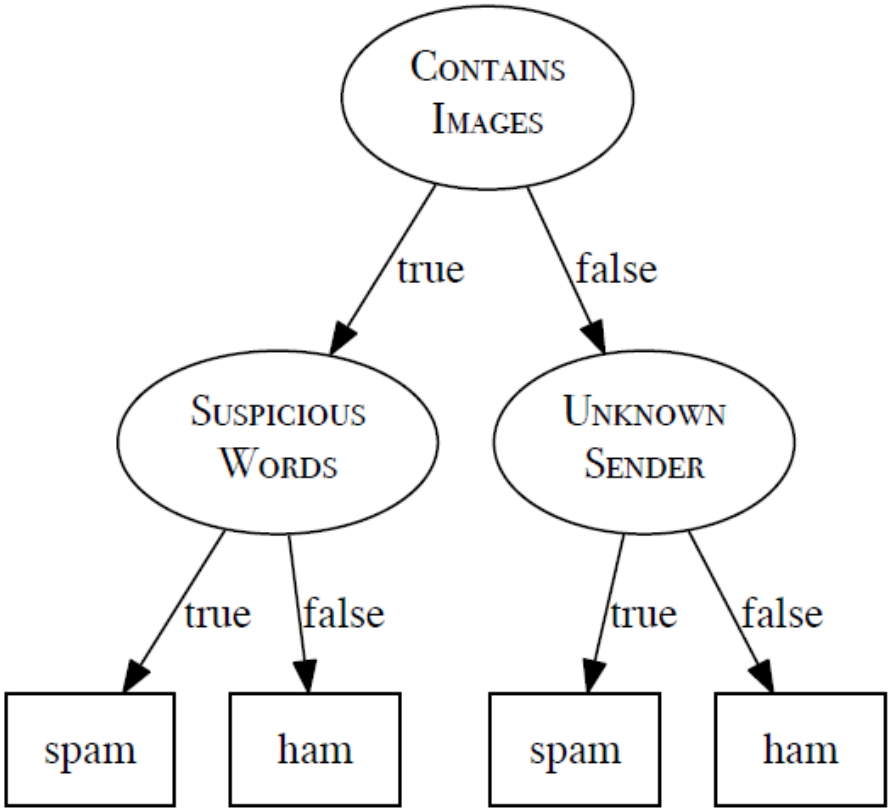


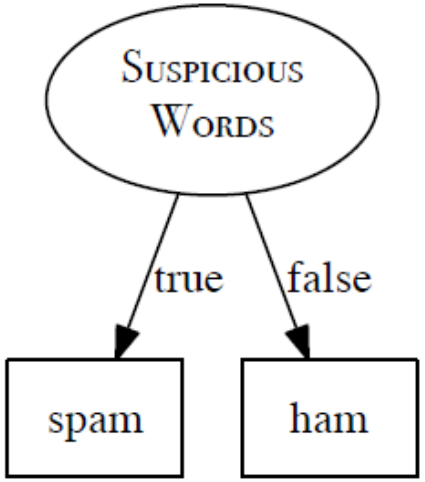
Suspicious Words	Unknown Sender	Contains Images	Class
true	false	true	spam
true	true	false	spam
true	true	false	spam

Suspicious Words	Unknown Sender	Contains Images	Class
false	true	true	ham
false	false	false	ham
false	false	false	ham

Structure of a Decision Tree

- Now consider the second shorter decision tree depicted on the right.
- Both of the decision trees below perfectly classify the training data.
- Which is preferable?





Structure of a Decision Tree

- "**Suspicious Words**" as a feature provides **more information (or provides a better insight)** into the value of a target class than the "**Contains Images**" feature.

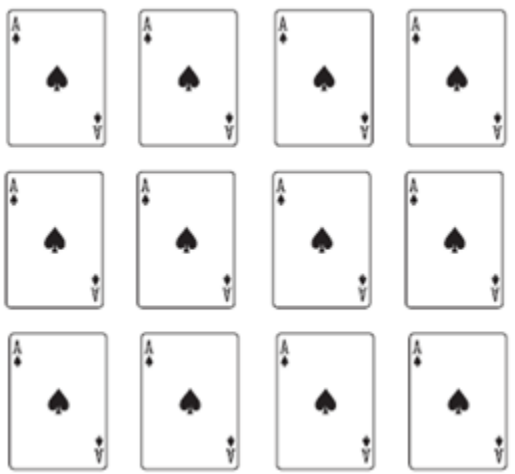
- Therefore, when building a decision tree we need some method for ranking or quantifying which features carry more information about the target.

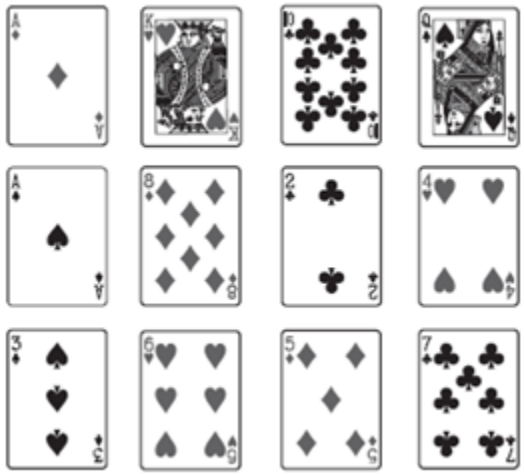
- A very common way of achieving this is called **Shannon's Entropy Model**.

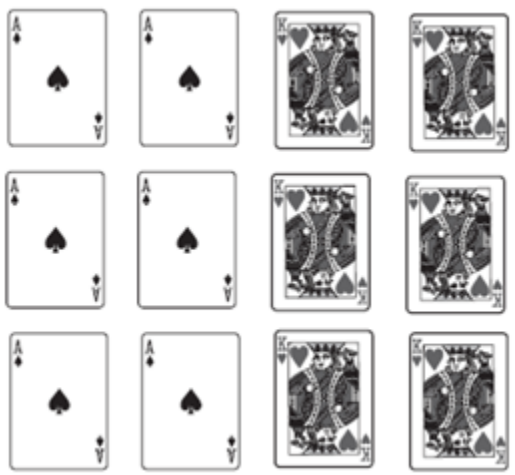
Shannon Entropy Metric

- Claude Shannon's entropy model defines a computational measure of the **impurity** of the elements of a set.

- An intuitive way to understand the entropy of a set is to think in terms of the **<u>uncertainty</u> associated with guessing the result if you were to make a random selection** from the set.
- An entropy value of zero means zero uncertainty, the higher the entropy value the higher the level of uncertainty.







Entropy

- Now lets consider entropy in the context of a simple dataset.
- What is the entropy of the Surf feature in the dataset below when Swell = Big.
  - Ent(Surf, Swell = Big) = **0**
- Every time the swell is big we go surf.
- The swell feature has two values. What it the entropy for **Ent(Surf, Swell = Small)**

Rain	Swell	Surf
Yes	Big	Yes
Yes	Small	No
No	Small	No
No	Big	Yes

Entropy

- The entropy for the Surf feature when Rain = Yes is high because there is a high level of uncertainty.
  - Ent(Surf, Rain = Yes) **= 1.**

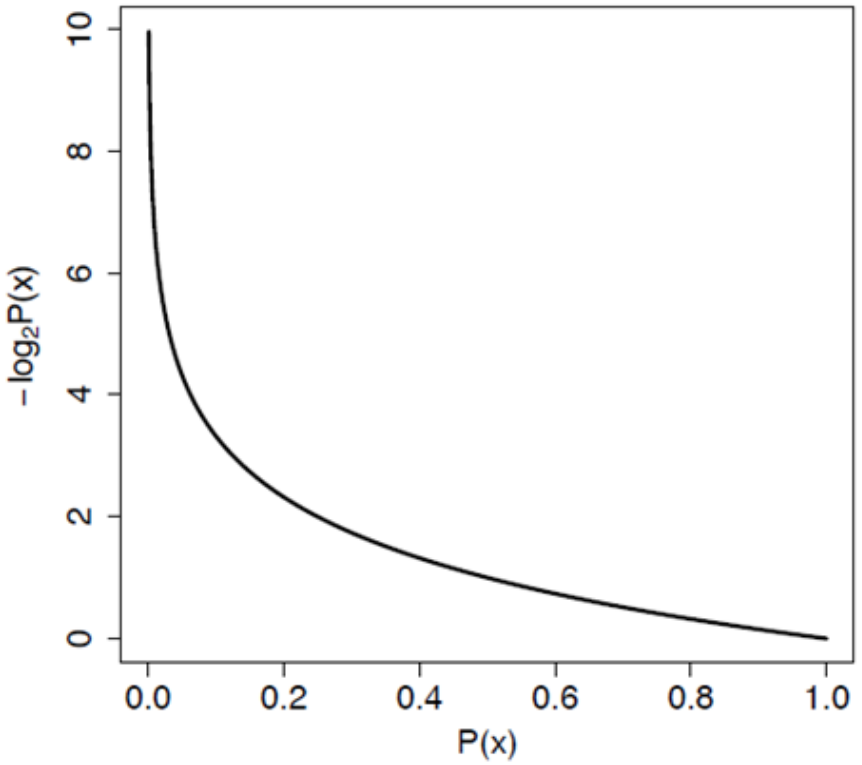
- In other words when it was raining, we went surfing on one occasion and once we didn't go surfing. Therefore the influence of this feature on whether or not we go surfing is highly uncertain.
- What is the entropy for the rain feature when it's value is No Ent(Surf, Rain='No').

Rain	Swell	Surf
Yes	Big	Yes
Yes	Small	No
No	Small	No
No	Big	Yes

Entropy

- Shannon's entropy is a **negative weighted sum** of the **logs** of the **probabilities** of each **possible outcome** when we make a random selection from a set.

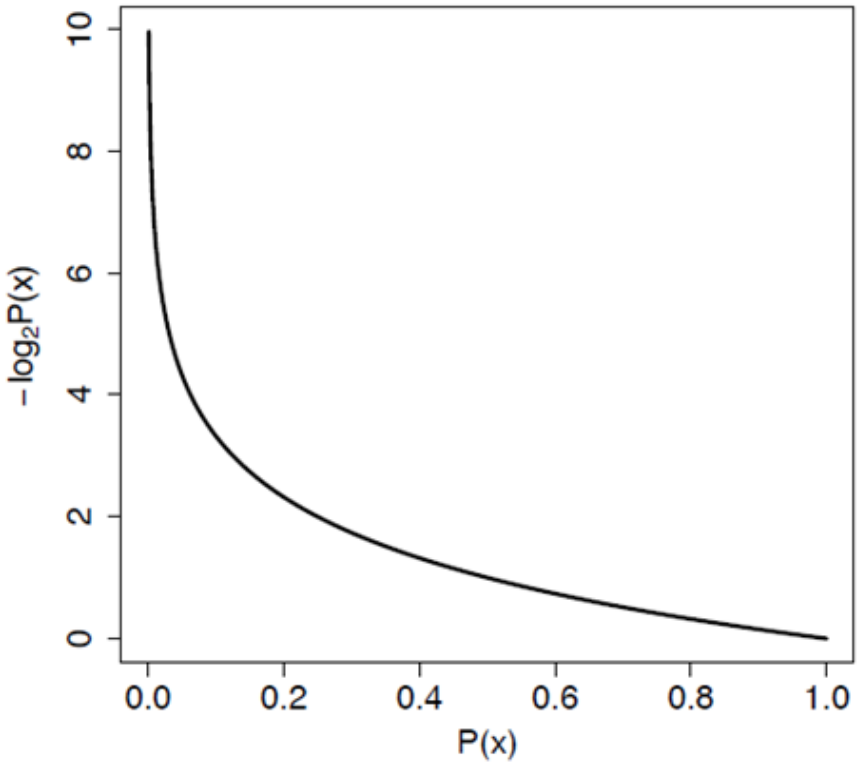
- The weights used in the sum are the probabilities of the outcomes themselves so that the outcomes with the higher probabilities contribute more to the overall entropy.



Entropy

- Shannon's entropy is a **negative weighted sum** of the **logs** of the **probabilities** of each **possible outcome** when we make a random selection from a set.

- The weights used in the sum are the probabilities of the outcomes themselves so that the outcomes with the higher probabilities contribute more to the overall entropy.



Entropy

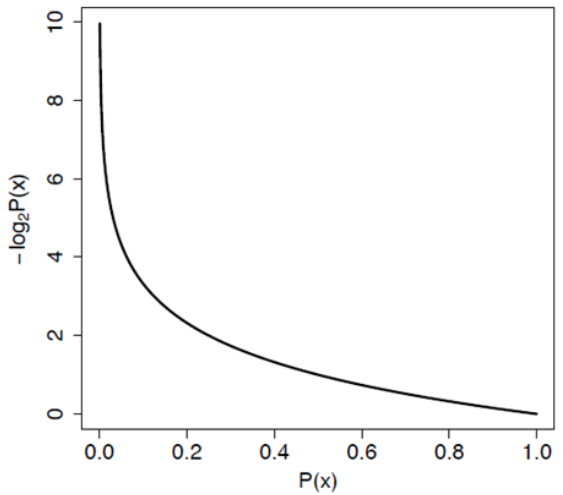
We can calculate the entropy of the sample S as:

$$\mathrm{Ent}(S) = \sum_{i=1}^{n} - p_i \log_2 p_i$$

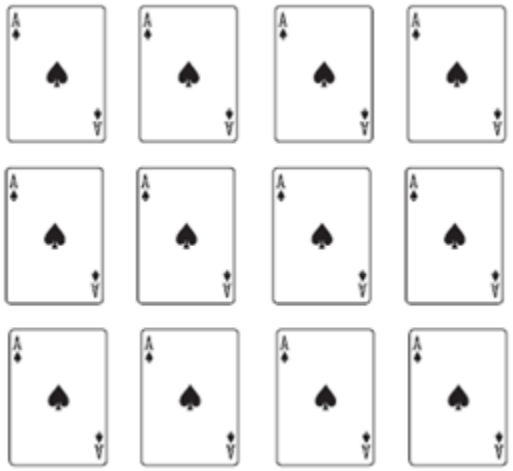
For a sample S with n different values, the probability of value i is p_i (number of instances of value i divided by total number of instances in the sample S)

Ent(s1) = -(12/12)$\log_2$ (12/12) = 0

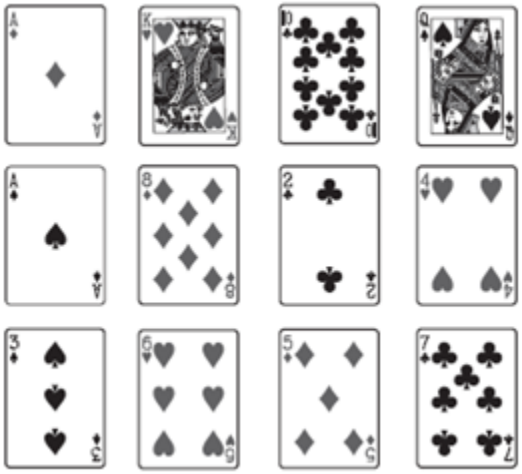
$$\text{Ent}(S) = \sum_{i=1}^{n} - p_i \log_2 p_i$$



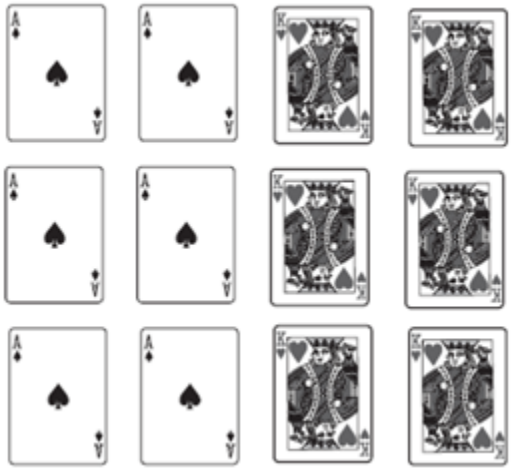
s1



s2

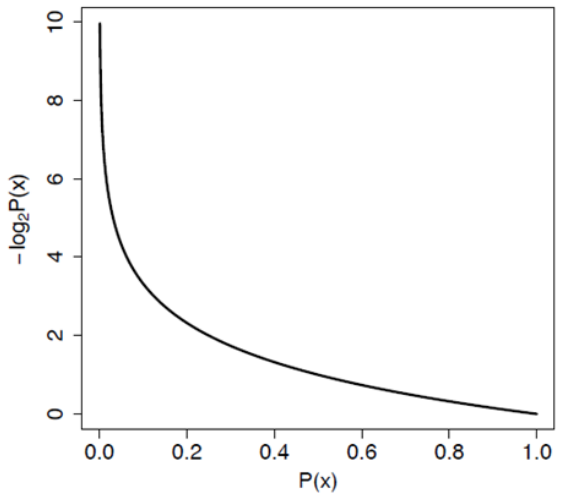


s3

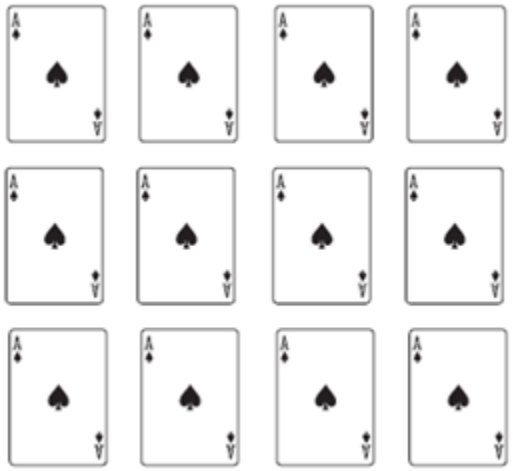


Ent(s2) = - (1/12)(log$_2$ (1/12)) * 12 = 3.58

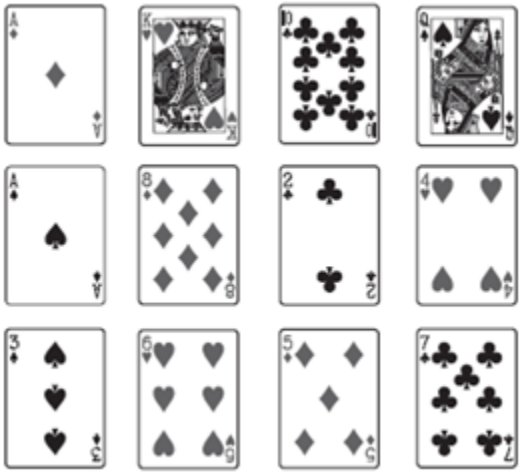
$$\text{Ent}(S) = \sum_{i=1}^{n} - p_i \log_2 p_i$$



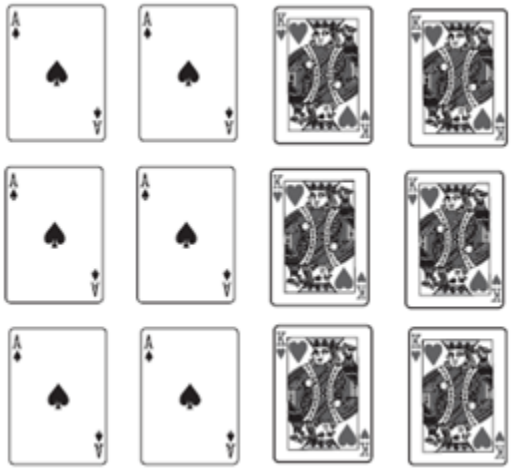
s1



s2

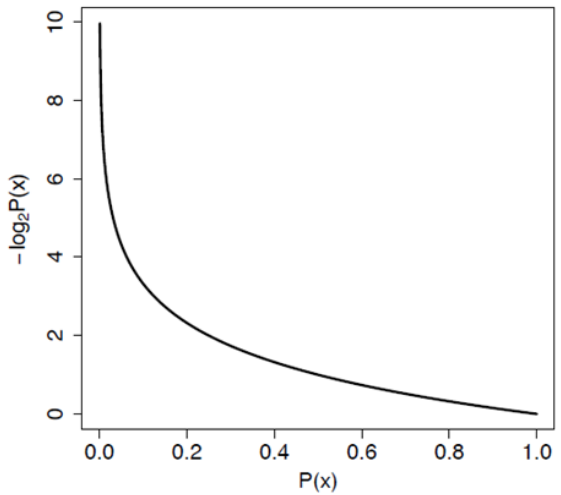


s3

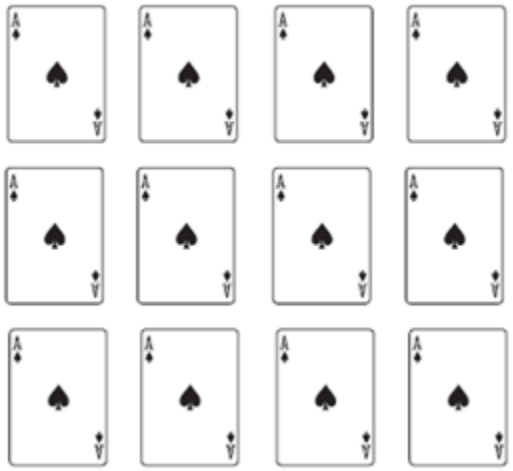


Ent(s3) = = - (6/12)($\log_2$(6/12)) - (6/12) ($\log_2$(6/12)) = 1

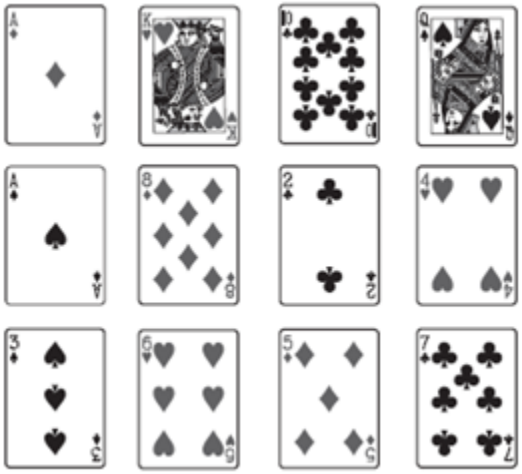
$$\text{Ent}(S) = \sum_{i=1}^{n} - p_i \log_2 p_i$$



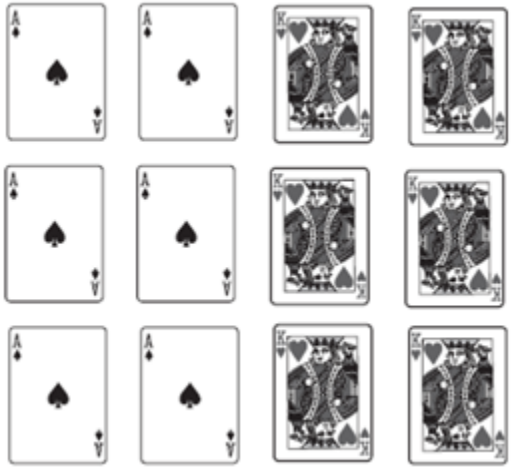
s1



s2



s3



Entropy Example

- In the example below I'm going to determine the entropy for the entire dataset S

$$\text{Ent}(S) = \sum_{i=1}^{n} - p_i \log_2 p_i$$

- There are two features in our dataset, therefore n=2.
- **Ent(Surf)** = ((-2/4) $\log_2$(2/4)) + ((-2/4) $\log_2$(2/4)) = 1

Rain	Swell	Surf
Yes	Big	Yes
Yes	Small	No
No	Small	No
No	Big	Yes

Entropy Example

$$\mathrm{Ent}(S) = \sum_{i=1}^{n} - p_i \log_2 p_i$$

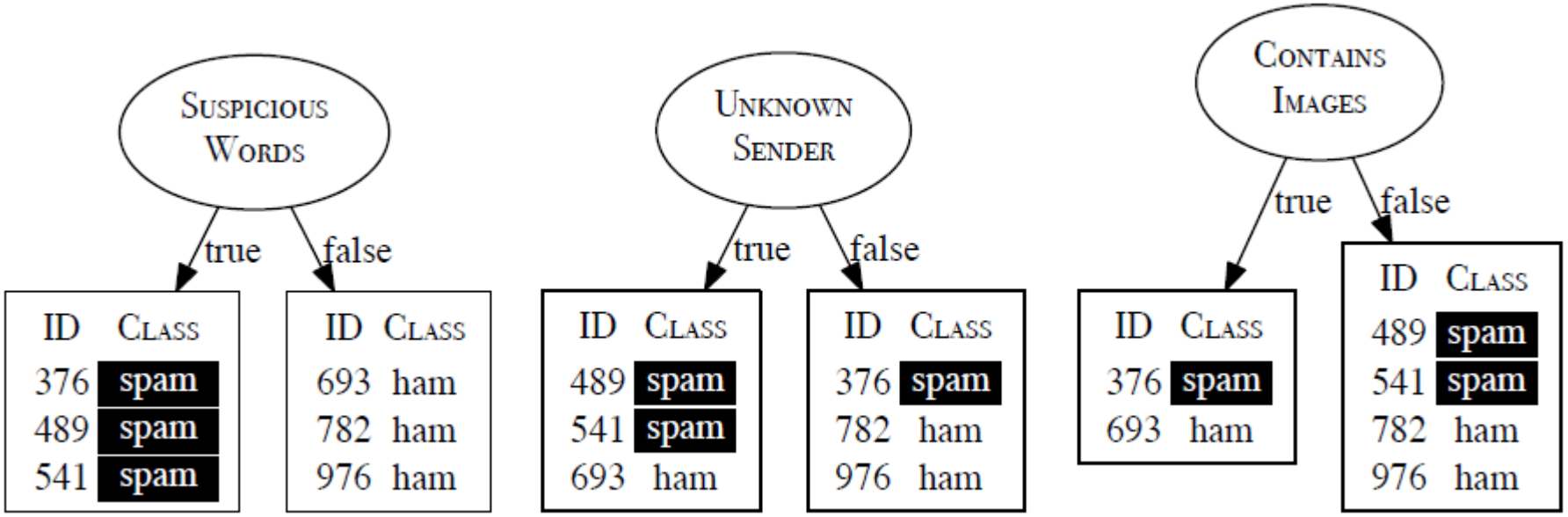
- Now lets examine the entropy of the swell feature when Swell = Big.
- **Ent(Surf, Swell=Big)** = (-2/2) $\log_2(2/2)$ = 0 (No uncertainty in this data)

Rain	Swell	Surf
Yes	Big	Yes
Yes	Small	No
No	Small	No
No	Big	Yes

Entropy

- So now that we understand entropy let's go back to the process of building a decision tree. How does entropy help us?

Suspicious Words	Unknown Sender	Contains Images	Class
true	false	true	spam
true	true	false	spam
true	true	false	spam
false	true	true	ham
false	false	false	ham
false	false	false	ham



Obtaining Information Gain

- A common method used to quantify the level of information carried by a feature for a specific class is called Information Gain.

- **Information Gain is a measure of the reduction in the overall entropy of a set of instances that is achieved by using a specific feature to split the data at a specific node.**

- Computing information gain is broken into three steps:
  - Compute the entropy of the **original sample**.
  - For each descriptive feature, create the sets that result by partitioning the instances in the sample using their feature values. Then **sum the weighted entropy scores** of each of these sets.
  - **Subtract the summed entropy values** after we have split the sample using this specific feature from the **original entropy** to give the information gain.

Information Gain

- Information Gain of an feature is the reduction in Entropy from partitioning the data according to values of that feature.

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

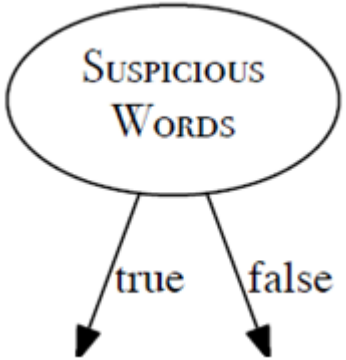
Subset of S where A has value v

Size of S

S is the incoming data, A is the specific feature. Above we add the weighted entropy for each possible value (v) of the feature A and subtract from the entropy of the original data sample

Suspicious Words	Unknown Sender	Contains Images	Class
true	false	true	spam
true	true	false	spam
true	true	false	spam
false	true	true	ham
false	false	false	ham
false	false	false	ham

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

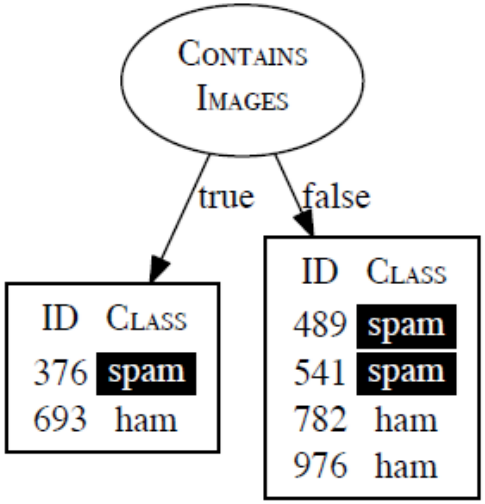


Suspicious Words	Unknown Sender	Contains Images	Class
true	false	true	spam
true	true	false	spam
true	true	false	spam

Suspicious Words	Unknown Sender	Contains Images	Class
false	true	true	ham
false	false	false	ham
false	false	false	ham

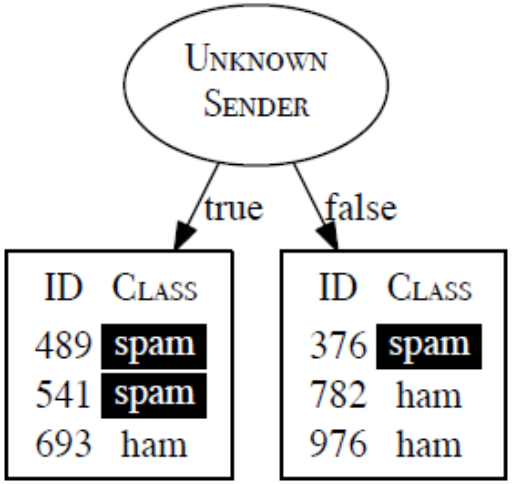
Suspicious Words	Unknown Sender	Contains Images	Class
true	false	true	spam
true	true	false	spam
true	true	false	spam
false	true	true	ham
false	false	false	ham
false	false	false	ham

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



Suspicious Words	Unknown Sender	Contains Images	Class
true	false	true	spam
true	true	false	spam
true	true	false	spam
false	true	true	ham
false	false	false	ham
false	false	false	ham

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



High Level Steps of Creating a Decision Tree

- High Level Steps for Creating a Decision Tree

1. Begin **with a dataset** containing all features and resulting classes
2. Find the **feature that best splits** the dataset class using information gain
3. Divide the data in groups based on the feature that you have used to split (**node**)
4. With **each group** find the feature to best **split the data** set class. Continue this process.