Confusion Matrix & Decision Trees

CONFUSION MATRIX

Although a confusion matrix provides the information needed to determine how well a classification model performs, summarizing this information with a single number would make it more convenient to compare the performance of different models. This can be done using a performance metric such as accuracy, which is defined as follows.

Table 4.2. Confusion matrix for a 2-class problem.

		Predicted Class	
		$Class = 1$	$Class = 0$
Actual Class	$Class = 1$	f_{11}	f_{10}
	$Class = 0$	f_{01}	f_{00}

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}. \quad (4.1)$$

Equivalently, the performance of a model can be expressed in terms of its **error rate**, which is given by the following equation:

$$\text{Error rate} = \frac{\text{Number of wrong predictions}}{\text{Total number of predictions}} = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}. \quad (4.2)$$

Most classification algorithms seek models that attain the highest accuracy, or equivalently, the lowest error rate when applied to the test set. We will revisit

HOW A DECISION TREE WORKS

In this instance we are classifying animals into mammals or non-mammals based off given attributes. The approach is to ask a series of questions that directs us to another question while eliminating options along the way.
For example, is the species cold or warm blooded? If it is cold blooded, this eliminates the mammal option. Otherwise, it is either a bird or a mammal. In the latter case, we need to ask a

follow up question: Do the females of the species give birth to their young? Those that do not give birth are definitely mammals, while those that do not are likely to be non-mammals (with the exception of egg laying mammals - Platypus).
We ask a series of carefully crafted questions about the attributes of the test records. Each time we receive an answer, a follow up question is asked until we reach a conclusion about the class label of the record. Decision trees are created through nodes and directed edges.

Root node - Has no incoming edges and zero or more outgoing edges. (The initial starting point)

Internal nodes - each of which has exactly one incoming edge and two or more outgoing edges.

Leaf or Terminal nodes - each of which has exactly one incoming edge and no outgoing edges.

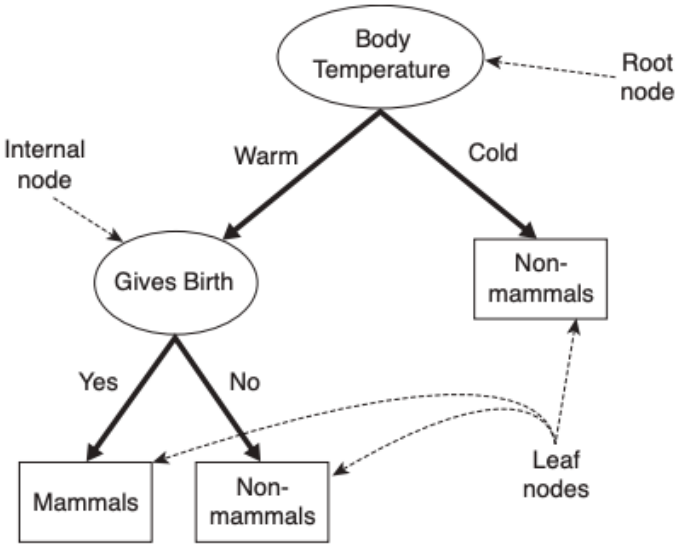
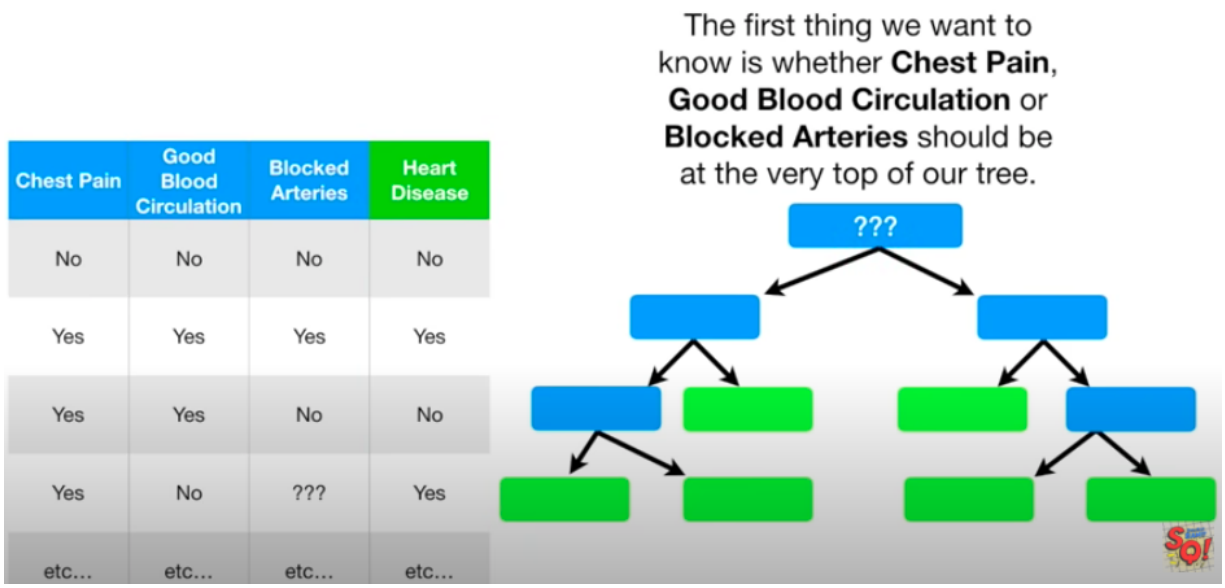


Figure 4.4. A decision tree for the mammal classification problem.

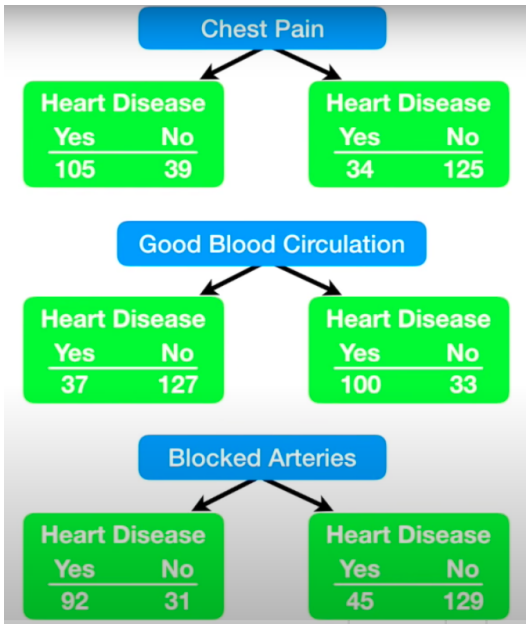
In a decision tree, each leaf node is assigned a class label. AKA the outcome of the prior node. All other nodes, such as root, and internal nodes contain test conditions to separate records that have different characteristics. I.E. Gives birth? Body Temperature?

For example, a flamingo has a warm body temperature, and does not give birth which results in a non-mammal classification.

HOW TO BUILD A DECISION TREE



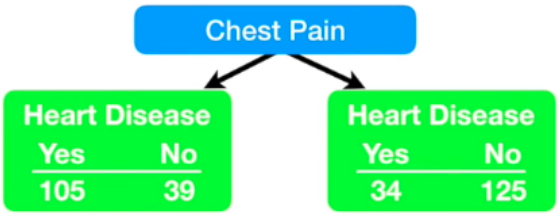
We start by looking at how well chest pain ALONE serves as a predictor for heart disease. Then we do the same for the remaining attributes - Good Blood Circulation & Blocked Arteries



** The goal is to decide whether chest pain, good blood circulation, or blocked arteries should be the first thing in our decision tree (aka the root node).

Because none of the leaf nodes are 100% “YES Heart Disease” or 100% “NO Heart Disease”, they are all considered “impure.” Therefore we need a way to measure impurity.

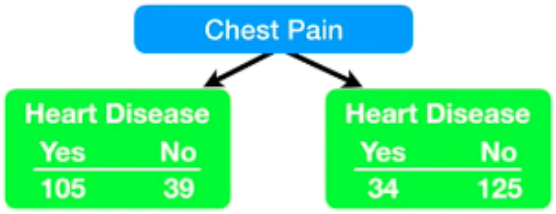
To do this we utilize Gini Impurity



For this leaf, the Gini impurity = 1 - (the probability of “yes”)^2 - (the probability of “no”)^2

$$= 1 - \left(\frac{105}{105 + 39}\right)^2 - \left(\frac{39}{105 + 39}\right)^2$$

$$= 0.395$$



Gini impurity = 0.395 0.336

Gini impurity for Chest Pain = weighted average of Gini impurities for the leaf nodes

$$= \left(\frac{144}{144 + 159}\right) 0.395 + \left(\frac{159}{144 + 159}\right) 0.336$$

$$= 0.364$$

After doing the Gini impurity for all three factors. Results are as follows:

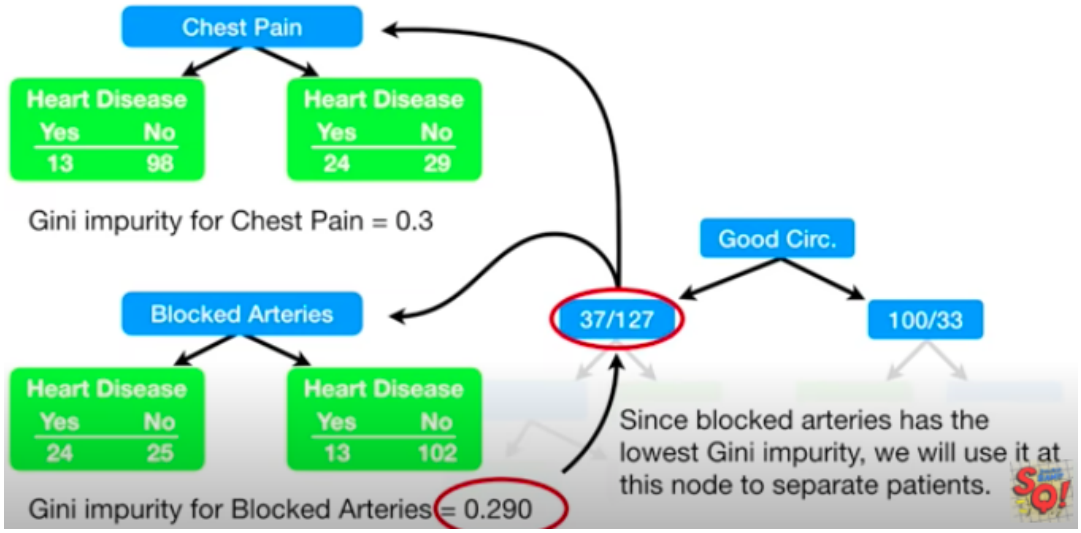
Chest Pain - .364 Good Blood Circulation - .360 Blocked Arteries - .381

Good blood circulation boasts the lowest Gini Impurity, therefore it separates patients with & without heart disease the best.

Therefore we use good blood circulation for our root node. Once we have our root node established, the process continues, and of the remaining attributes we look to see what has the lowest Gini impurity in classifying our patients correctly.

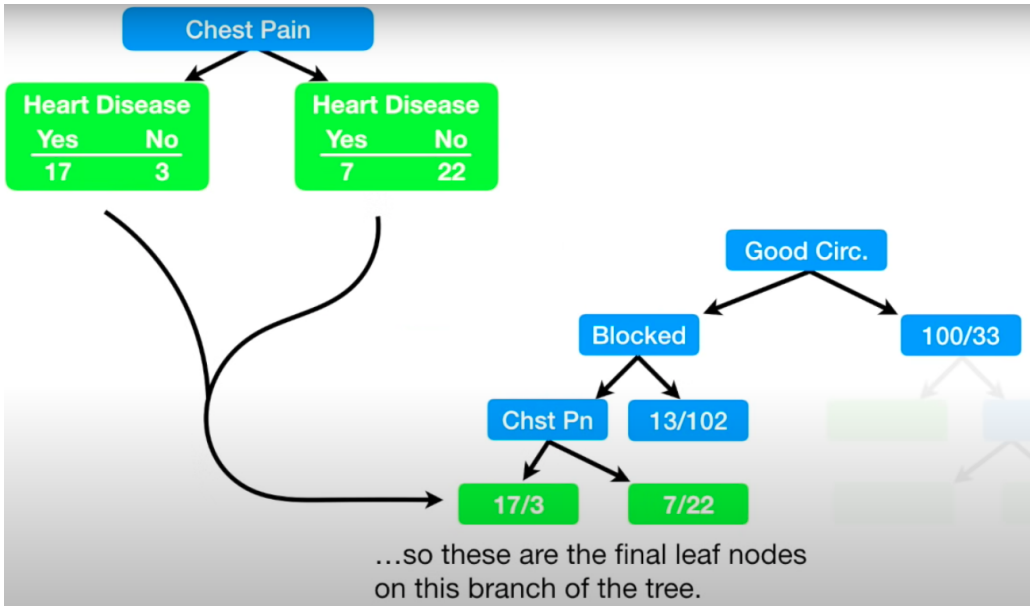
For the 164 patients on the left, Gini impurity for Blocked Arteries is .29 whereas Gini impurity for Chest Pain is .30

*37/127 & 100/33 is Y/N for heart disease

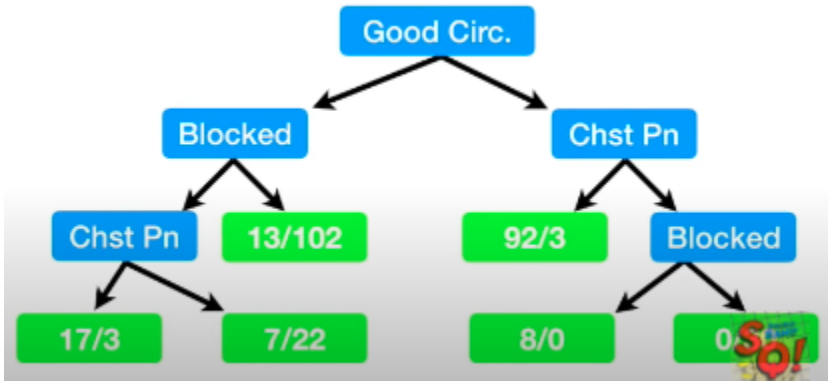


To recap, we started at the top by separating patients with good circulation. Then we used blocked arteries to separate patients on the left side of the tree. All we have left is chest pain, so we'll see how well it separates the

BUILDING A DECISION TREE WITH NUMERIC DATA



When we use chest pain to divide the 115 patients on the right side (13/102), the Gini impurity comes out to 0.29. The original gini impurity of that given node by separating on blocked arteries is 0.2, so we come to the conclusion that we can stick to that as our final leaf node. This allows us to finalize the left side of the tree, and then we repeat the process on the right side.

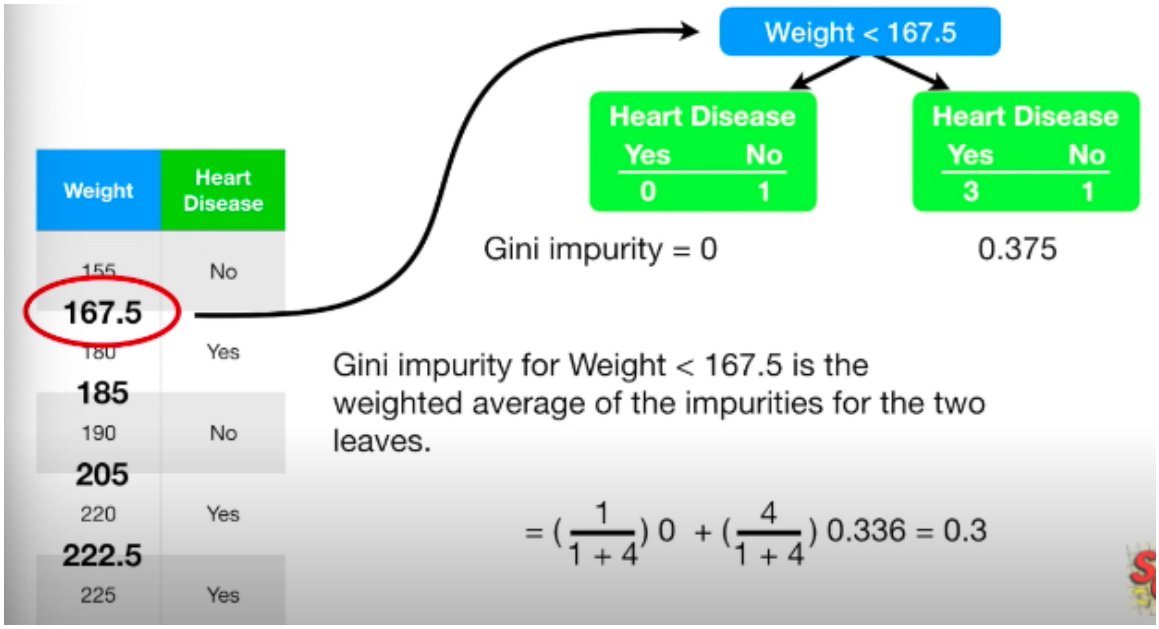


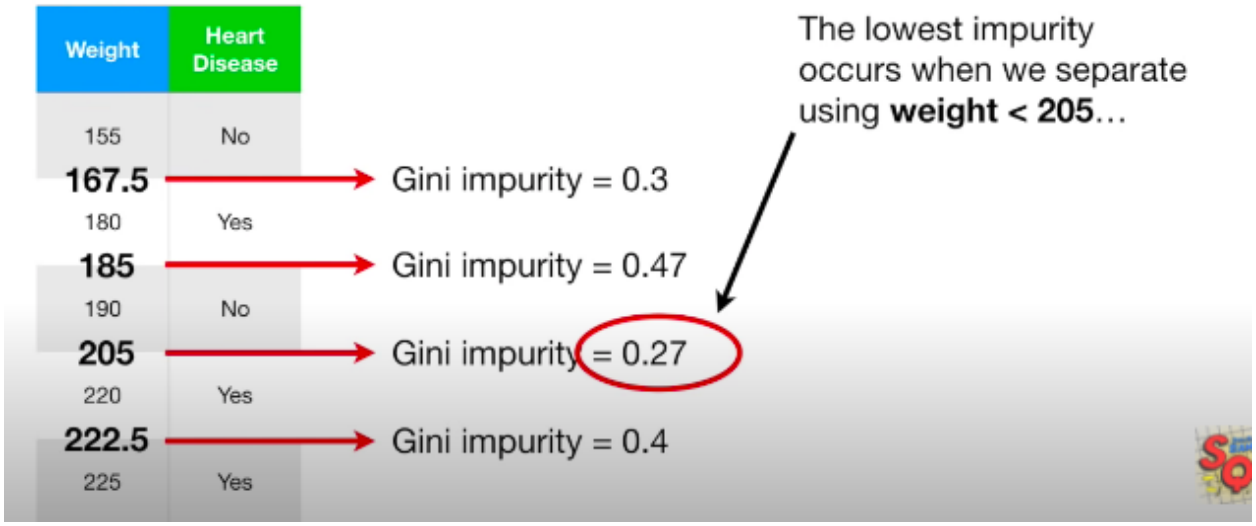
- 1) Calculate all of the Gini impurity scores
- 2) If the node itself has the lowest score, then there is no point in separating the patients any more and it becomes a leaf node.
- 3) If separating the data results in an improvement, then pick the separating with the lowest impurity value.

BUILDING A DECISION TREE WITH NUMERIC DATA

Step 1 - Sort the patients by weight, lowest to highest.
Step 2 - Calculate the average weight for all adjacent patients
Step 3 - Calculate the impurity values for each average weight

Example - Calculate Gini impurity for weight less than 167.5.





205 is the cutoff and impurity value that we will use when we compare weight to chest pain or blocked arteries.

WHAT IS THE ALTERNATIVE TO GINI IMPURITY - ENTROPY

- Entropy is a measure of disorder in a dataset.
- Information gain is a measure of the decrease in disorder achieved by partitioning on an attribute in the dataset
- Entropy is always a calculation on a vector of categorical variable values.

Entropy ($E(\vec{d})$)

$$E(\vec{d}) = -\sum_{i=1}^{k} p_i \log_2(p_i)$$

Entropy for a Partition

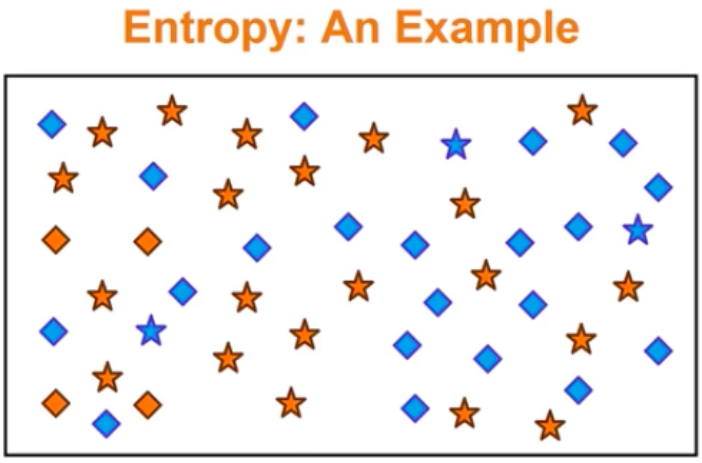
$$E(\vec{d}, \vec{a}) = \sum_{j=1}^{m} \left(\frac{n_j}{n} \times E(\vec{d}_j) \right)$$

Information Gain ($I(\vec{d}, \vec{a})$)

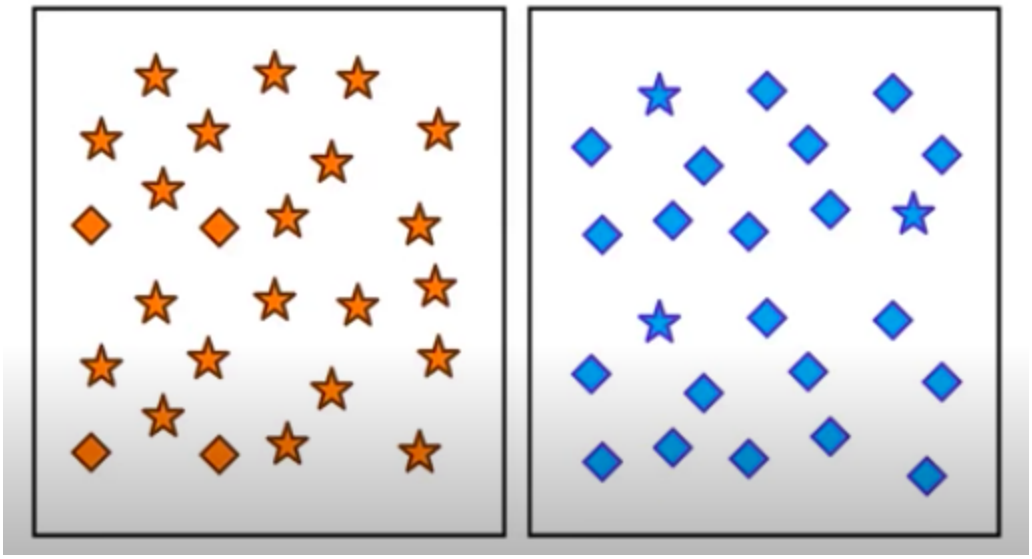
$$I(\vec{d}, \vec{a}) = E(\vec{d}) - E(\vec{d}, \vec{a})$$

EXAMPLE

The target attribute is star vs diamond. We are interested in predicting if we are to reach into the box if we choose star vs diamond. We have a rough split with 49 objects in the box, 25 stars, and 24 diamonds. We also have color codes, so not all diamonds are blue, and not all stars are orange. So let's go ahead and partition on color



Entropy: An Example



When we partition on color there is a lot less disorder here, and greater likelihood that we can expect a star or a diamond depending on the box.

25 orange objects, 21 stars, 4 diamonds

24 blue objects, 3 stars, 21 diamonds

Full Group:

$$E(\vec{d}) = -\sum_i p_i \log_2(p_i)$$
$$= -(p_1 \log_2(p_1) + p_2 \log_2(p_2))$$
$$= -\left(\frac{24}{49}\log_2\left(\frac{24}{49}\right) + \frac{25}{49}\log_2\left(\frac{25}{49}\right)\right)$$
$$\cong 0.9997$$

Orange Group:

$$E(\vec{d}, orange) = -\sum_i p_i \log_2(p_i)$$
$$= -(p_1 \log_2(p_1) + p_2 \log_2(p_2))$$
$$= -\left(\frac{4}{25}\log_2\left(\frac{4}{25}\right) + \frac{21}{25}\log_2\left(\frac{21}{25}\right)\right)$$
$$\cong 0.6343$$

Entropy score turns out to be .6343 is quite a bit lower than the full group, which means a lot less disorder. If you reach into the box knowing it is orange, and you predict it is likely to be a star you have a much better chance of being correct in your prediction.

Blue Group:

$$E(\vec{d}, blue) = -\sum_i p_i \log_2(p_i)$$
$$= -(p_1 \log_2(p_1) + p_2 \log_2(p_2))$$
$$= -\left(\frac{21}{24}\log_2\left(\frac{21}{24}\right) + \frac{3}{24}\log_2\left(\frac{3}{24}\right)\right)$$
$$\cong 0.5436$$

There is even less disorder in the blue box. We have one less object here so an even greater chance of predicting correctly.

Combined Entropy:

$$E(\vec{d}) = 0.9997$$
$$E(\vec{d}, \vec{a}) = \frac{25}{49}E(orange) + \frac{24}{49}E(blue)$$
$$= \frac{25}{49}(0.6343) + \frac{24}{49}(0.5436) = 0.5899$$

Information Gain:

$$I(\vec{d}, \vec{a}) = E(\vec{d}) - E(\vec{d}, \vec{a}) = 0.9997 - 0.5899 = 0.4097$$

We take total objects in the orange group times the entropy, and do the same for the blue group. Once we have that answer it is passed into the information gain formula where we take the original entropy of the entire group, and subtract the weighted entropy.
Partitioning on color allows us this information gain because it makes it much easier for us to predict whether it'll be a star or diamond.

Real world scenario - Jury Selection, we have parameters such as M/F, religion, occupation. What we could ahead of time is a survey in the community, and see whether they would be leaning guilty not guilty for people of similar backgrounds and classify accordingly.

What if we had a box that was all stars and no diamonds?

Entropy $(E(\vec{d}))$

$$E(\vec{d}) = -\sum_{i=1}^{k} p_i \log_2(p_i)$$

What happens if we have a term with $0 \log_2(0)$?

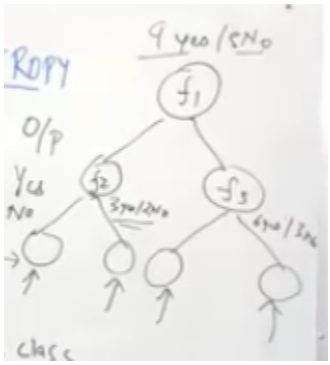
By definition, we set:

$$0 \log_2(0) = 0$$

ENTROPY IN DECISION TREE INTUITION

The first step in splitting a decision tree is to split on the right attribute. In order to measure this we use entropy as a measurement.

Given F1 as the root node, and F2 & F3 as the internal nodes, we are looking to measure the purity of the split on F1.



We have a total of 9 yes and 5 no in our decision tree, when we split off into F2 we have 3 yes and 2 no. Lastly in our F3 split we have 6 yes and 3 no. Totaling 15 results

Given these results we must measure the success via entropy, in order to choose what node we want to go with.

Keep in mind a pure split would be a clear decision all yes or all nos.

Example 4 yes/ 0 nos = 0 bit

Example 3 yes/ 3 nos = 1 bit

WE WANT ENTROPY as LOW AS POSSIBLE.

HISHAM - Entropy is a level of uncertainty.

When we have 6, 5 the uncertainty of which class is high.
When we have 1, 2 the uncertainty of which class is still high.
When we have 2, 2 we have peak uncertainty at 1.
When we have 5, 0 we have absolutely 0 uncertainty.

```
from math import log

def entropy(*probs):

  try:
    total = sum(probs)
    return sum([-p / total * log(p / total, 2) for p in probs])
  except:
    return 0

print(entropy(6, 5), entropy(1, 2), entropy(2, 2), entropy (9,5), entropy (5,0))
```

```
0.9940302114769565 0.9182958340544896 1.0 0.9402859586706309 0
```

Here is an example of deciding what parameter we want to initially split on.
Outlook boasts the greatest information gain which is subtracted from the original dataset entropy.

The goal is to always reach leaves where certainty is maximized, and entropy is minimized.

ISSUES

The main issue with decision trees is that they tend to overfit. A way to fix that issue is via pruning. Pruning is known as: