Decision Trees

IE 506 Lecture

March 12, 2024

1. Classification Algorithms
   - Decision Trees

Classification Algorithms: Decision Trees

Decision Tree

Dataset 1:

Body Temperature (°F)	Visit to Foreign Countries	Antibodies in blood	Disease Presence
100	NO	NO	NO
98	YES	NO	NO
102	YES	NO	NO
104	YES	YES	YES
99	YES	YES	YES
100	NO	YES	YES

- Rows denote **samples**.
- Last column denotes the **output** (or response or dependent) variable or **label**.
- First three columns denote **attributes** (also called features).
- **NOTE:** There are only two output values YES and NO for Disease Presence. (recall e-mail spam classification)

Decision Tree

Dataset 1:

Body Temperature (°F)	Visit to Foreign Countries	Antibodies in blood	Disease Presence
100	NO	NO	NO
98	YES	NO	NO
102	YES	NO	NO
104	YES	YES	YES
99	YES	YES	YES
100	NO	YES	YES

- Body Temperature attribute takes continuous values. Hence Body Temperature is called **continuous** attribute.
- Visit to Foreign Countries and Antibodies in blood take only two values. Hence Visit to Foreign Countries and Antibodies in Blood are called **binary** attribute.
- There are other types of attributes: Nominal, Categorical etc. for which we will see some examples later.

Decision Tree

Dataset 1:

Body Temperature (°F)	Visit to Foreign Countries	Antibodies in blood	Disease Presence
100	NO	NO	NO
98	YES	NO	NO
102	YES	NO	NO
104	YES	YES	YES
99	YES	YES	YES
100	NO	YES	YES

- **Aim 1:** To learn a classification machine learning model on Dataset 1 using the first three columns of the samples as features and the last column as the output label.

Decision Tree

Dataset 1:

Body Temperature (°F)	Visit to Foreign Countries	Antibodies in blood	Disease Presence
100	NO	NO	NO
98	YES	NO	NO
102	YES	NO	NO
104	YES	YES	YES
99	YES	YES	YES
100	NO	YES	YES

- **Aim 1:** To learn a classification machine learning model on Dataset 1 using the first three columns of the samples as features and the last column as the output label.
- **Aim 2:** Use the learned model to find the status of Disease Presence for a new sample with the attributes Body Temperature, Visit to Foreign Countries and Antibodies in Blood.

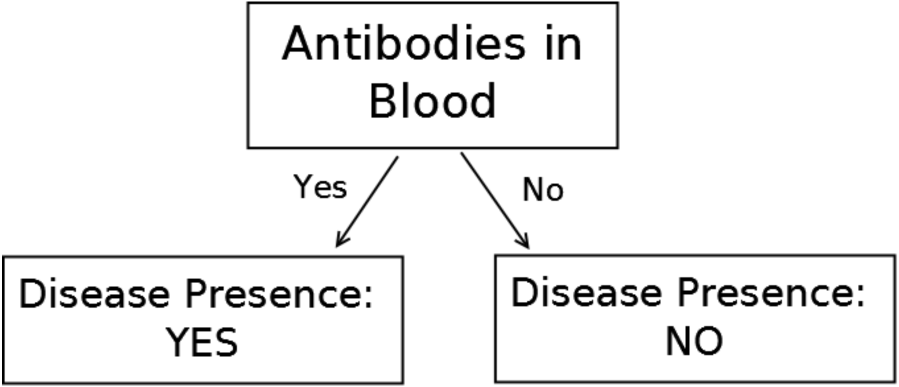
Decision Tree

Dataset 1:

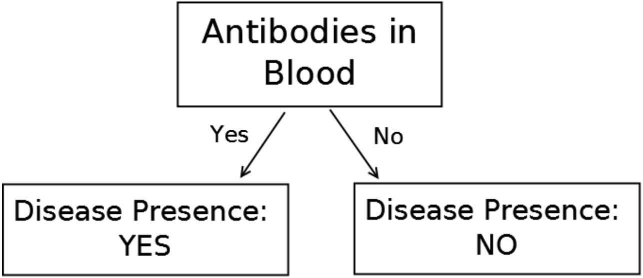
Body Temperature (°F)	Visit to Foreign Countries	Antibodies in blood	Disease Presence
100	NO	NO	NO
98	YES	NO	NO
102	YES	NO	NO
104	YES	YES	YES
99	YES	YES	YES
100	NO	YES	YES

- **NOTE:** The Antibodies in Blood feature is perfectly correlated with Disease Presence.
- Hence for Dataset 1, it would be simply possible to indicate Disease Presence just by knowing the status of Antibodies in Blood.

Decision Tree For Dataset 1



Decision Tree For Dataset 1



- So if we have trained using dataset 1, our decision tree finds the status of Disease Presence by simply checking Antibodies in Blood.
- Hence if a new sample is to be tested, the decision tree will examine only the Antibodies in Blood attribute of the new sample and decide Disease Presence accordingly.

Decision Tree

Dataset 2:

Body Temperature (°F)	Visit to Foreign Countries	Antibodies in blood	Disease Presence
100	NO	NO	NO
98	YES	NO	YES
102	YES	NO	NO
104	YES	YES	YES
99	YES	YES	NO
100	NO	YES	YES

- **NOTE:** No feature is perfectly correlated with the Disease Presence output.
- **Question:** How do we construct a decision tree now?

Decision Tree

Dataset 2:

Body Temperature (°F)	Visit to Foreign Countries	Antibodies in blood	Disease Presence
100	NO	NO	NO
98	YES	NO	YES
102	YES	NO	NO
104	YES	YES	YES
99	YES	YES	NO
100	NO	YES	YES

- **Question:** How do we construct a decision tree now?
- We will start with the simpler case: Let us ignore Body Temperature attribute for the time being and consider only the attributes Visit to Foreign Countries and Antibodies in blood.

Decision Tree Construction for Dataset 2

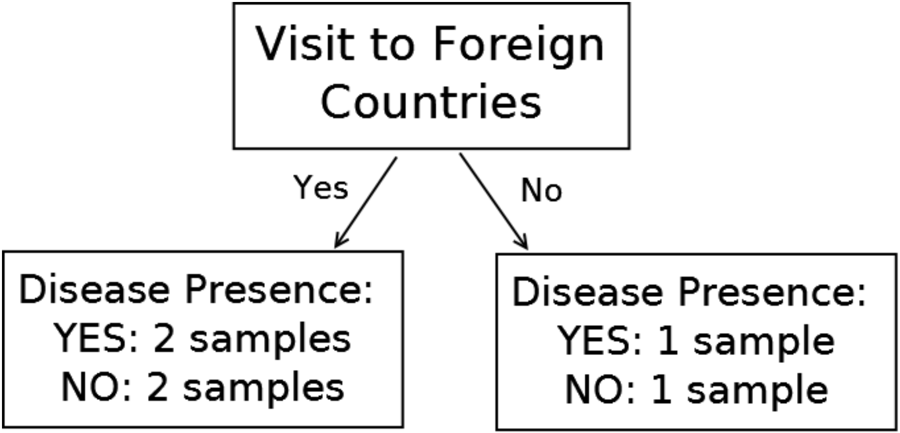
- Let us check how the splits look when we split on the Visit to Foreign Countries attribute.

Dataset 2:

Body Temperature (°F)	Visit to Foreign Countries	Antibodies in blood	Disease Presence
100	NO	NO	NO
98	YES	NO	YES
102	YES	NO	NO
104	YES	YES	YES
99	YES	YES	NO
100	NO	YES	YES

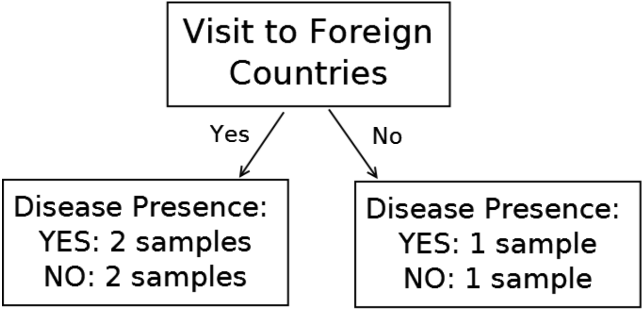
Decision Tree Construction for Dataset 2

- Splitting on the Visit to Foreign Countries attribute we have:



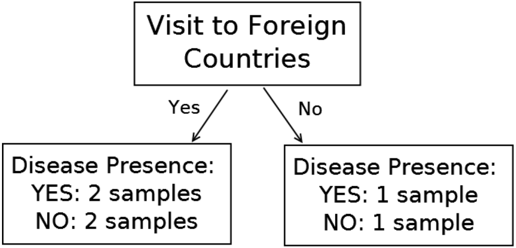
Decision Tree Construction for Dataset 2

- Splitting on the Visit to Foreign Countries attribute we have:



- Notice that the split produces two **nodes** corresponding to Visit to Foreign Countries=YES and Visit to Foreign Countries=NO.

Decision Tree Construction for Dataset 2



- We see that among those who visited foreign countries, 50% have disease and 50% do not have disease.
- Similarly, among those who did not visit foreign countries, 50% have disease and 50% do not have disease.
- Thus vaguely, just by knowing the status of Visit to Foreign Countries attribute, we can only be 50% sure that the person has a disease.

Decision Tree Construction for Dataset 2

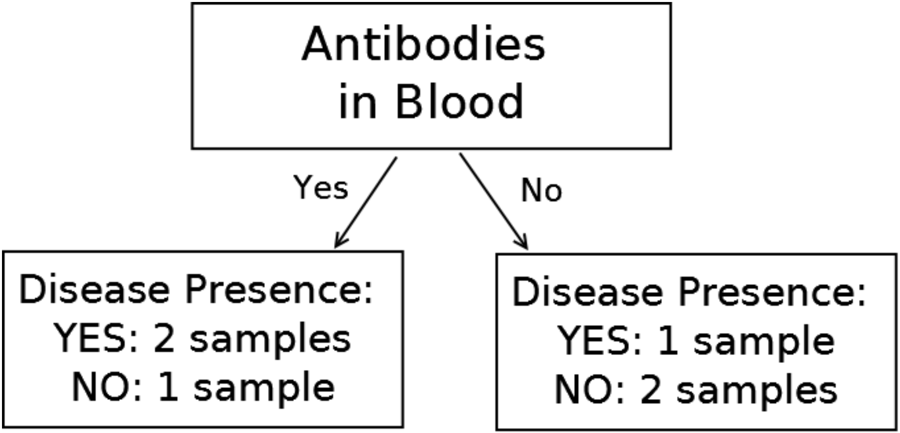
- Let us check how the splits look when we split on the Antibodies in Blood attribute.

Dataset 2:

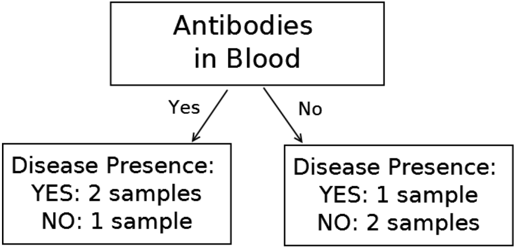
Body Temperature (°F)	Visit to Foreign Countries	Antibodies in blood	Disease Presence
100	NO	NO	NO
98	YES	NO	YES
102	YES	NO	NO
104	YES	YES	YES
99	YES	YES	NO
100	NO	YES	YES

Decision Tree Construction for Dataset 2

- Splitting on the Antibodies in Blood attribute we have:

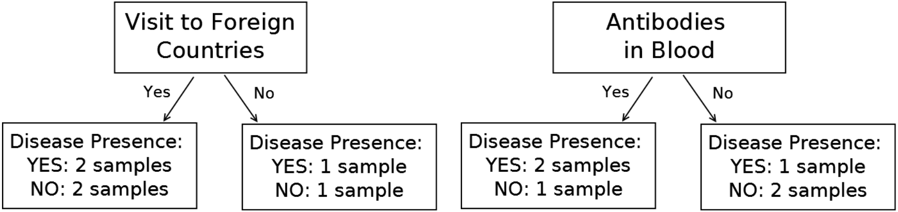


Decision Tree Construction for Dataset 2



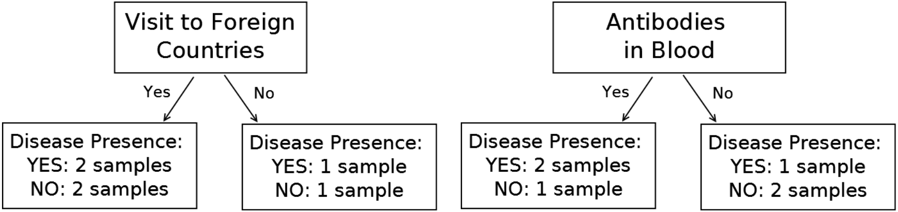
- We see that among those who have antibodies in blood, 66.6% have disease and 33.3% do not have disease.
- Among those who do not have antibodies in blood, 66.6% do not have disease and 33.3% have disease.
- Thus vaguely, just by knowing the status of Antibodies in Blood attribute, we can say with $>$50% confidence that the person has a disease.

Decision Tree Construction for Dataset 2



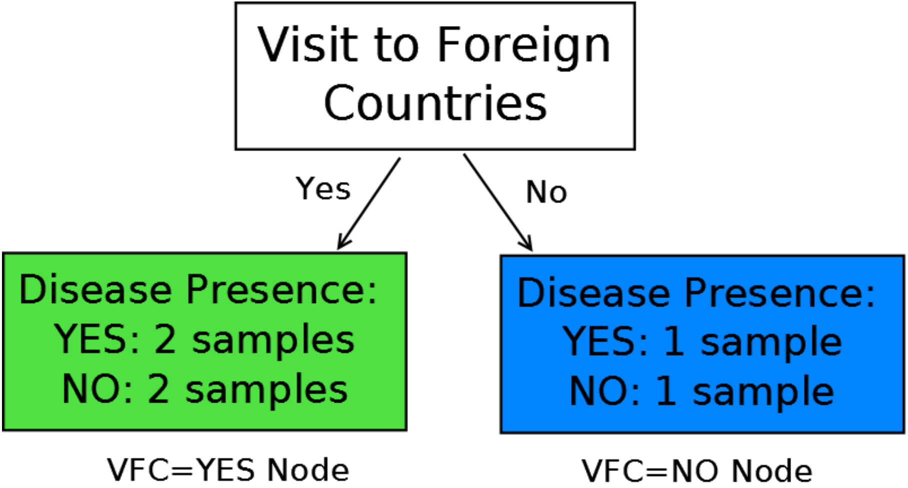
- Thus, splitting using Antibodies in Blood attribute produces somewhat better confidence in classifying the Disease Presence label when compared to Antibodies in Blood attribute.
- Let us now make this intuition more formal.

Decision Tree Construction for Dataset 2



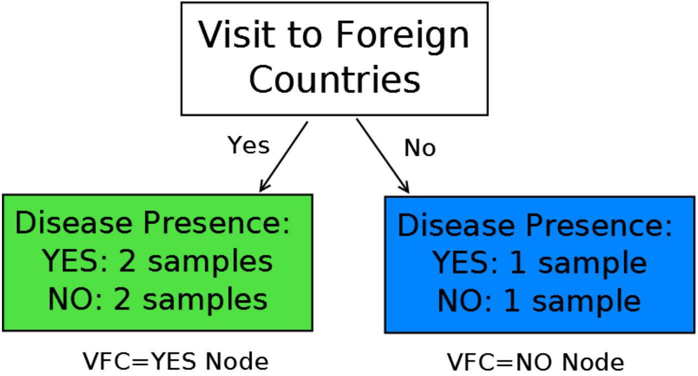
- Let us denote Disease Presence using **DP**, Visit to Foreign Countries as **VFC** and Antibodies in Blood using **AB**.

Decision Tree Construction for Dataset 2



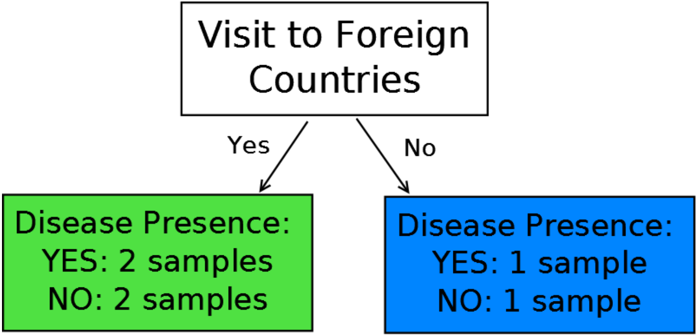
- Notice (and recall) that the split produces VFC=YES node (in the left) and VFC=NO node (in the right).

Decision Tree Construction for Dataset 2



- Note that there are 4 samples at VFC=YES node.
- Now the probability that DP is YES given that VFC is YES is given by: $P(DP = YES|VFC = YES) = 2/4 = 0.5$.
- We can immediately derive that $P(DP = NO|VFC = YES) = 1 - P(DP = YES|VFC = YES) = 1 - 0.5 = 0.5$.

Decision Tree Construction for Dataset 2

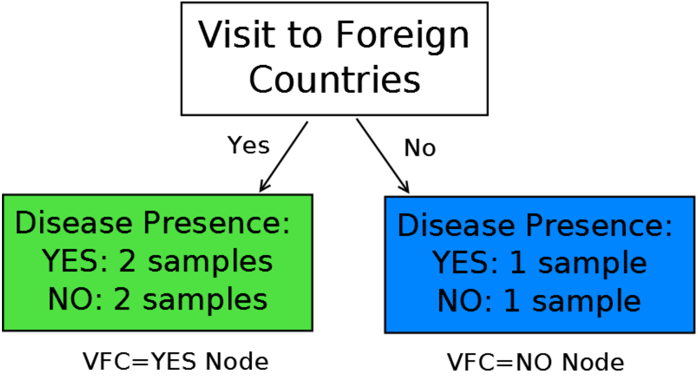


VFC=YES Node

VFC=NO Node

- Note that there are 2 samples at VFC=NO node.
- The probability that DP is YES given that VFC is NO is given by: $P(DP = YES|VFC = NO) = 1/2 = 0.5$.
- Hence $P(DP = NO|VFC = NO) = 1 - P(DP = YES|VFC = NO) = 0.5$.

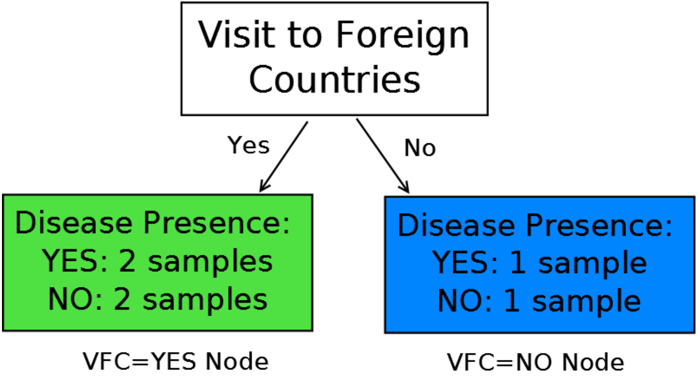
Decision Tree Construction for Dataset 2



- **Definition**: We define Entropy of the node VFC=YES as:

$$\text{Entropy(VFC=YES Node)} = \\ -P(DP = YES|VFC = YES)\log_2 P(DP = YES|VFC = YES) \\ -P(DP = NO|VFC = YES)\log_2 P(DP = NO|VFC = YES).$$

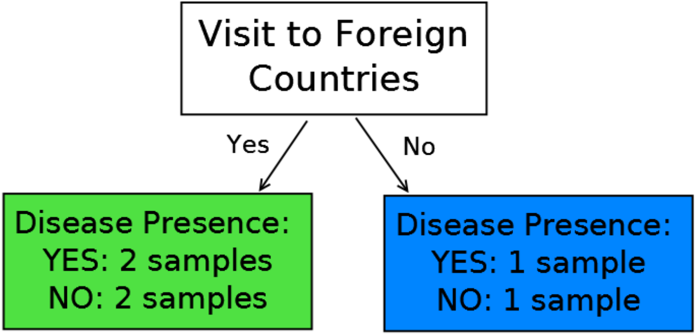
Decision Tree Construction for Dataset 2



- **Definition**: We define Entropy of the node VFC=NO as:

$$\text{Entropy(VFC=NO Node)} =$$
$$-P(DP = YES|VFC = NO)\log_2 P(DP = YES|VFC = NO)$$
$$-P(DP = NO|VFC = NO)\log_2 P(DP = NO|VFC = NO).$$

Decision Tree Construction for Dataset 2



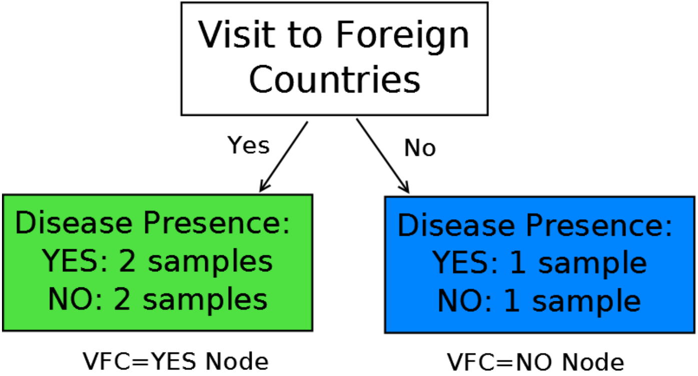
VFC=YES Node

Entropy:
-P(DP=YES|VFC=YES) log P(DP=YES|VFC=YES)
-P(DP=NO|VFC=YES) log P(DP=NO|VFC=YES)

VFC=NO Node

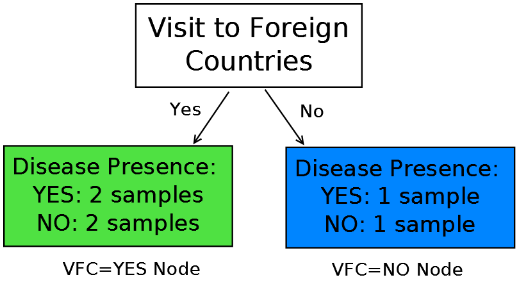
Entropy:
-P(DP=YES|VFC=NO) log P(DP=YES|VFC=NO)
-P(DP=NO|VFC=NO) log P(DP=NO|VFC=NO)

Decision Tree Construction for Dataset 2



- The Entropy value measures the level of **impurity** of a node.
- By impurity, we mean in some sense the amount of confusion present in a node to declare the output value Disease Presence as YES or NO.
- Hence lower impurity value $\implies$ low confusion in deciding the output value.

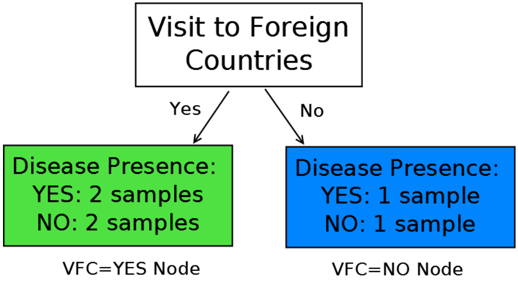
Decision Tree Construction for Dataset 2



- We can now compute Entropy of the node VFC=YES as:

$$
\begin{aligned}
&\text{Entropy(VFC=YES Node)} = \\
&-P(DP = YES|VFC = YES)\log_2 P(DP = YES|VFC = YES) \\
&-P(DP = NO|VFC = YES)\log_2 P(DP = NO|VFC = YES). \\
&= -0.5\log_2 0.5 - 0.5\log_2 0.5 = -\log_2 0.5 = 1
\end{aligned}
$$

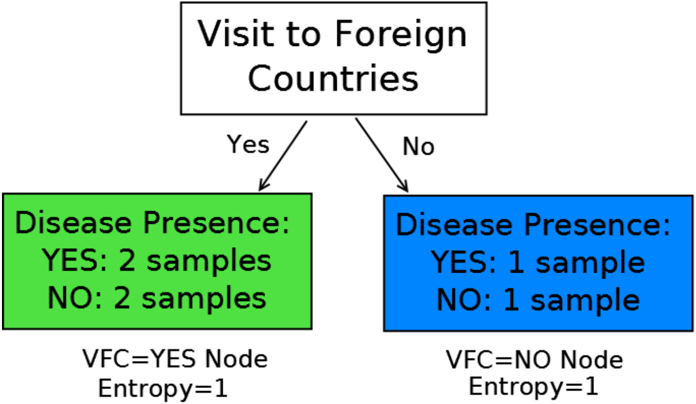
Decision Tree Construction for Dataset 2



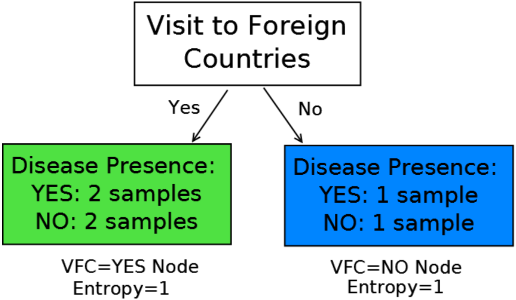
- Similarly, we can compute Entropy of the node VFC=NO is:

$$\begin{aligned}
&\text{Entropy(VFC=NO Node)} = \\
&- P(DP = YES|VFC = NO)\log_2 P(DP = YES|VFC = NO) \\
&- P(DP = NO|VFC = NO)\log_2 P(DP = NO|VFC = NO) \\
&= -0.5\log_2 0.5 - 0.5\log_2 0.5 = -\log_2 0.5 = 1
\end{aligned}$$

Decision Tree Construction for Dataset 2

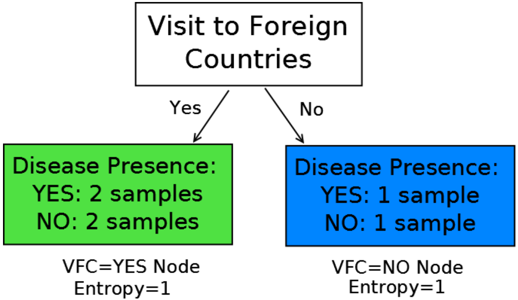


Decision Tree Construction for Dataset 2



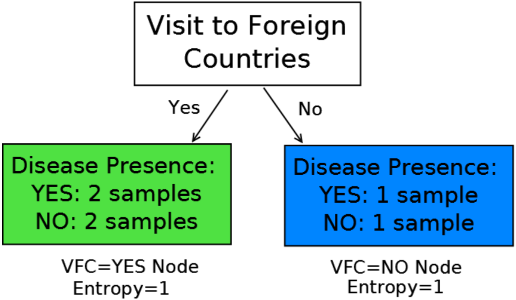
- What is special about the Entropy?

Decision Tree Construction for Dataset 2



- What is special about the Entropy?
- Note: The probabilities $P(DP = YES|VFC = YES)$ and $P(DP = No|VFC = YES)$ can be represented as p_1 and $1 - p_1$.

Decision Tree Construction for Dataset 2



- What is special about the Entropy?
- Note: The probabilities $P(DP = YES|VFC = YES)$ and $P(DP = NO|VFC = YES)$ can be represented as p_1 and $1 - p_1$.
- Hence

$$\text{Entropy(VFC=YES)} = -p_1 \log_2 p_1 - (1 - p_1)\log_2(1 - p_1)$$

Decision Tree Construction for Dataset 2

Consider

$$\text{Entropy(VFC=YES)} = -p_1 \log_2 p_1 - (1 - p_1) \log_2 (1 - p_1)$$

- When $p_1 = 1$ or $p_1 = 0$ the Entropy value is 0.
- When p_1 is 0.5 the Entropy is 1.
- Thus when we are sure about an event (indicated by $p_1 = 0$ and $p_1 = 1$), the entropy has a low value.
- Thus when we are not sure (or confused) about an event (indicated by $p_1 = 0.5$), the entropy has a high value.

Decision Tree Construction for Dataset 2

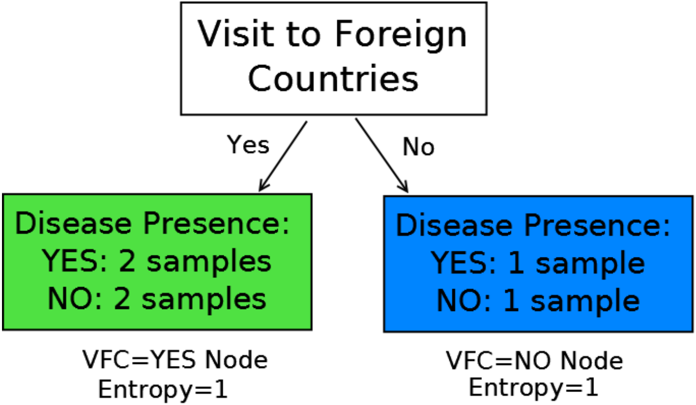
- Thus Entropy is a formal notion for the level of confusion in decision making process.
- High Entropy $\implies$ High confusion $\implies$ Cannot decide for sure.
- Low Entropy $\implies$ Low confusion $\implies$ Decision can be done with high confidence.

Decision Tree Construction for Dataset 2

Equivalently:

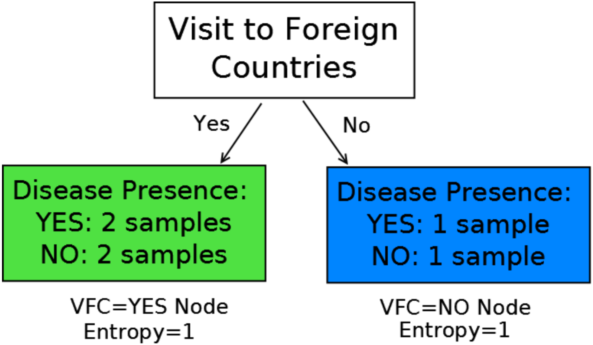
- High Entropy $\implies$ High Impurity $\implies$ High confusion $\implies$ Cannot decide for sure.
- Low Entropy $\implies$ Low Impurity $\implies$ Low confusion $\implies$ Decision can be done with high confidence.

Decision Tree Construction for Dataset 2



- Note that before the split, there were 6 samples in total.
- After splitting using VFC, 4 samples have moved to VFC=YES node and 2 samples have moved to VFC=NO node.

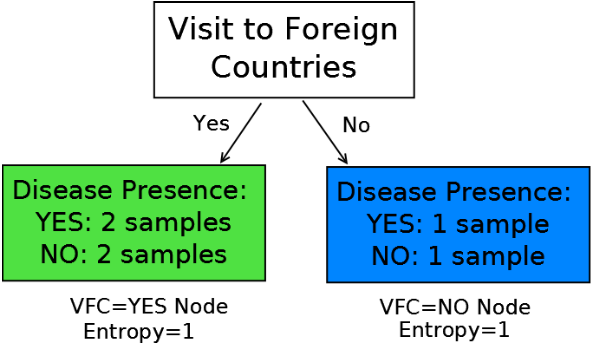
Decision Tree Construction for Dataset 2



- Now we can compute the **weighted impurity** associated with the VFC attribute as:

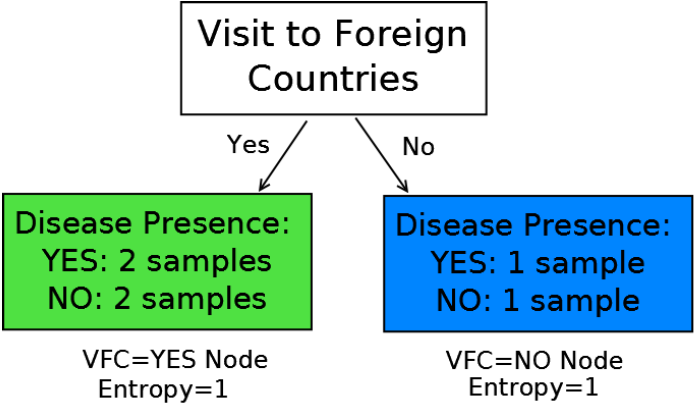
$$
\begin{aligned}
I(VFC) &= 4/6 * Entropy(VFC = YES) + 2/6 * Entropy(VFC = NO) \\
&= 4/6 * 1 + 2/6 * 1 = 4/6 + 2/6 = 1.
\end{aligned}
$$

Decision Tree Construction for Dataset 2



- **NOTE: Weighted impurity** is associated with an attribute whereas Entropy is associated with a node.

Decision Tree Construction for Dataset 2

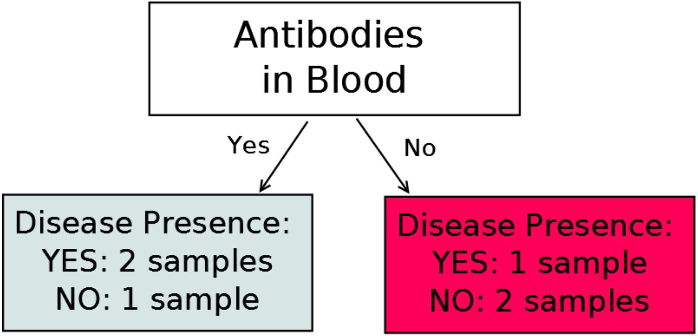


- Again we would want the weighted impurity associated with an attribute to be as small as possible.
- Low weighted impurity $\implies$ Low confusion in deciding output.

Decision Tree Construction for Dataset 2

- We will now repeat the calculations for the Antibodies in Blood attribute and compute the weighted impurity.

Decision Tree Construction for Dataset 2

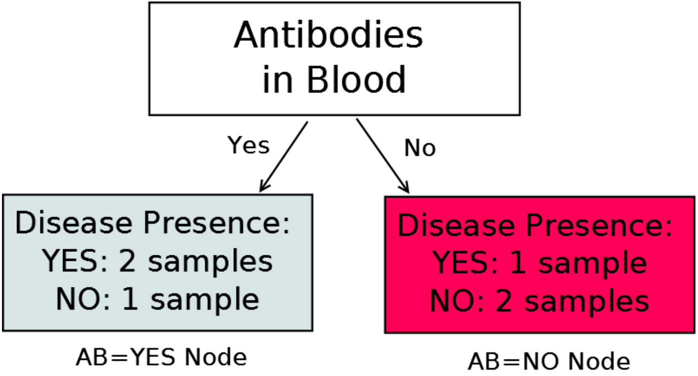


AB=YES Node

AB=NO Node

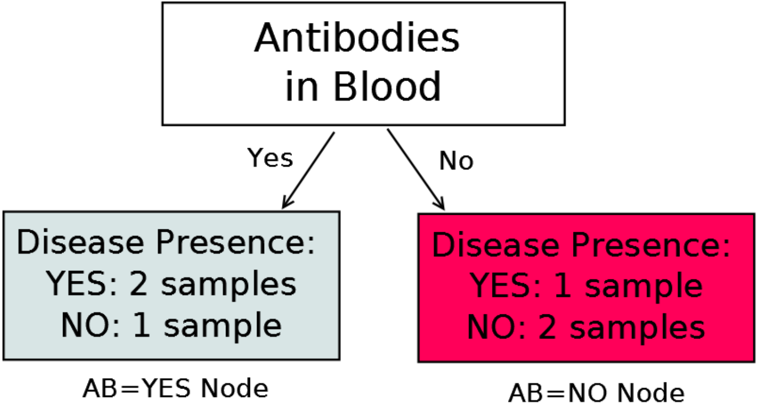
- Note that there are 3 samples at AB=YES node.
- Now the probability that DP is YES given that AB is YES is given by: $P(DP = YES|AB = YES) = 2/3 \approx 0.67$.
- We can immediately derive that $P(DP = NO|AB = YES) = 1 - P(DP = YES|AB = YES) = 1 - 0.67 = 0.33$.

Decision Tree Construction for Dataset 2



- Also note there are 3 samples at AB=NO node.
- Now the probability that DP is NO given that AB is NO is given by: $P(DP = NO|AB = NO) = 2/3 \approx 0.67$.
- We can immediately derive that $P(DP = YES|AB = NO) = 1 - P(DP = NO|AB = NO) = 1 - 0.67 = 0.33$.

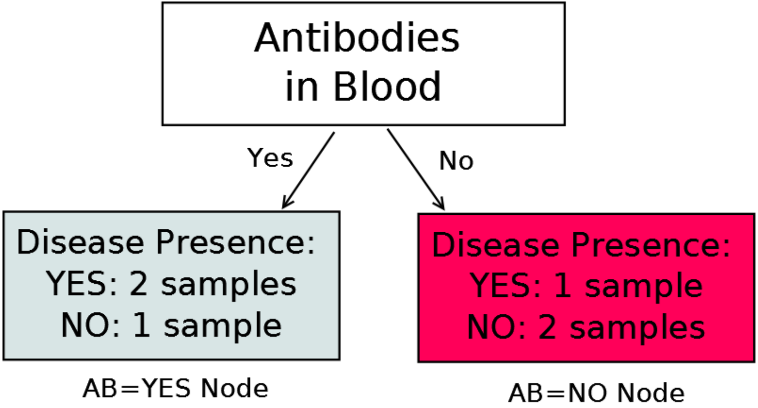
Decision Tree Construction for Dataset 2



- Hence we can compute the entropy for AB=YES node as:

$$\text{Entropy(AB=YES Node)} = ??$$

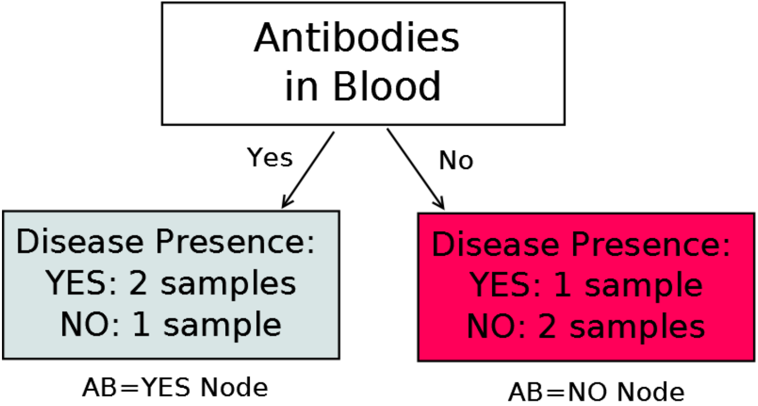
Decision Tree Construction for Dataset 2



- Hence we can compute the entropy for AB=YES node as:

$$\text{Entropy(AB=YES Node)} = 0.914926$$

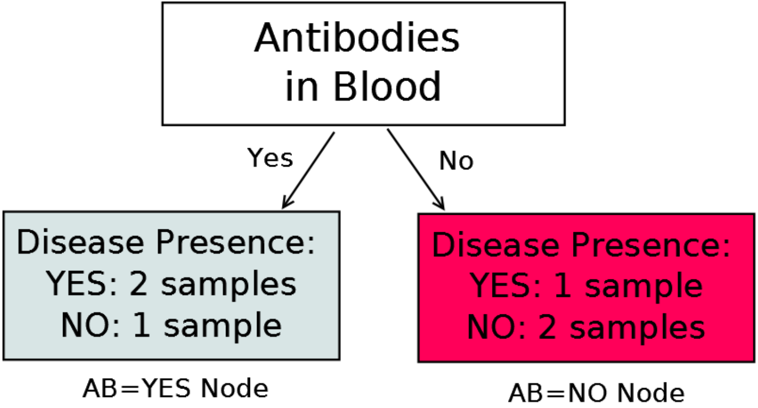
Decision Tree Construction for Dataset 2



- Also we can compute the entropy for AB=NO node as:

$$\text{Entropy}(\text{AB=NO Node}) = ??$$

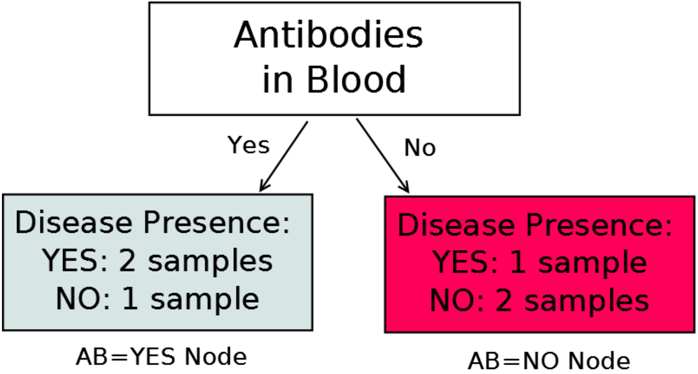
Decision Tree Construction for Dataset 2



- Also we can compute the entropy for AB=NO node as:

$$\text{Entropy(AB=NO Node)} = 0.914926$$

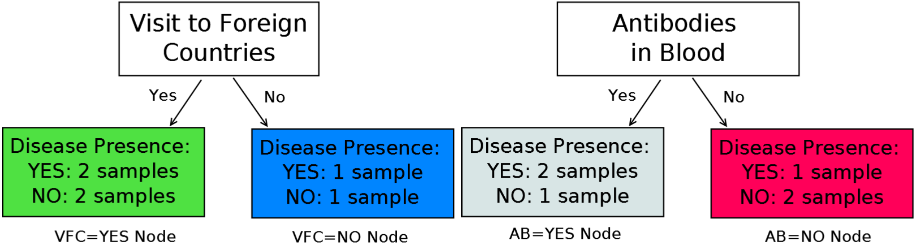
Decision Tree Construction for Dataset 2



- Before the split there are 6 samples.
- Note that during the split, 3 samples are at AB=YES node and 3 samples are at AB=NO node.
- So weighted impurity for antibodies in blood attribute is:

$$I(AB) = ??$$

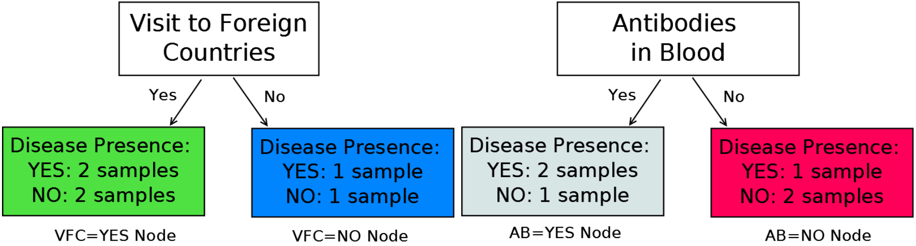
Decision Tree Construction for Dataset 2



- We have thus computed weighted impurity of VFC and AB as:

$$I(VFC) = 1$$
$$I(AB) = ??$$

Decision Tree Construction for Dataset 2



- We have thus computed weighted impurity of VFC and AB as:

$$I(VFC) = 1$$
$$I(AB) = ??$$

- **Note:** We need to choose that attribute which has a lower value of weighted impurity.

Decision Tree Construction for Dataset 2

- If $I(AB) < I(VFC)$ we need to choose Antibodies in blood.
- After splitting on Antibodies in blood attribute, we will have two partitions of the dataset:

Dataset 2 Split on AB attribute:

Body Temperature (°F)	Visit to Foreign Countries	Antibodies in blood	Disease Presence
104	YES	YES	YES
99	YES	YES	NO
100	NO	YES	YES

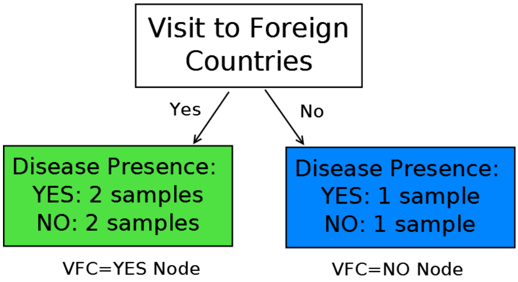
Body Temperature (°F)	Visit to Foreign Countries	Antibodies in blood	Disease Presence
100	NO	NO	NO
98	YES	NO	YES
102	YES	NO	NO

Decision Tree Construction for Dataset 2

- We need to repeat the split procedure for each of the partitions.

Decision Tree Construction for Dataset 2

Other notions of impurity

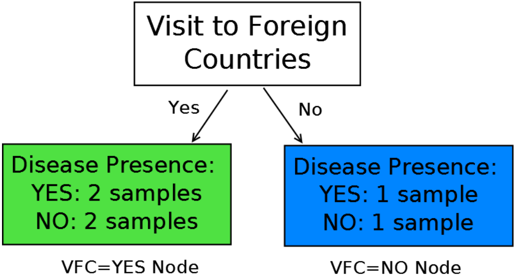


- We can compute Gini index of the node VFC=YES as:

$$\text{Gini(VFC=YES Node)} =$$
$$1 - [P(DP = YES|VFC = YES)]^2 - [P(DP = NO|VFC = YES)]^2$$
$$= 1 - 0.5^2 - 0.5^2 = 0.5$$

Decision Tree Construction for Dataset 2

Other notions of impurity



- Similarly, we can compute Gini index of the node VFC=NO as:

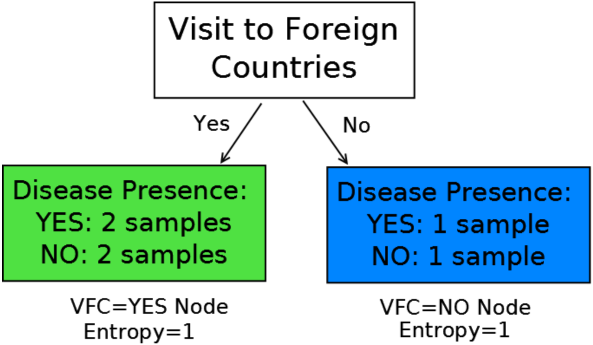
$$\text{Gini(VFC=NO Node)} =$$

$$1 - [P(DP = YES|VFC = NO)]^2 - [P(DP = NO|VFC = NO)]^2$$

$$= 1 - 0.5^2 - 0.5^2 = 0.5$$

Decision Tree Construction for Dataset 2

Other notions of impurity

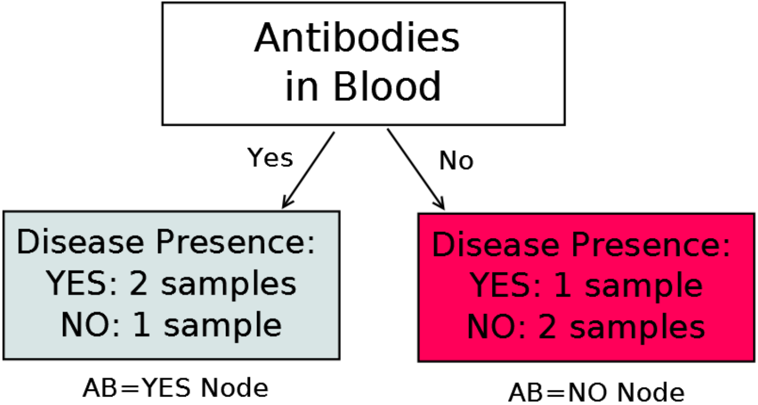


- Now we can compute the **weighted impurity** associated with the VFC attribute as:

$$\begin{aligned} I_{Gini}(VFC) &= 4/6 * Gini(VFC = YES) + 2/6 * Gini(VFC = NO) \\ &= 4/6 * 0.5 + 2/6 * 0.5 = (4/6 + 2/6) * 0.5 = 0.5. \end{aligned}$$

Decision Tree Construction for Dataset 2

Other notions of impurity



- **Exercise:** Compute the gini index for AB=YES and AB=NO node, and hence compute the weighted impurity $I_{Gini}(AB)$ associated with Antibodies in Blood attribute.

Decision Tree Construction for Dataset 2

- Note that we almost forgot Body Temperature attribute.

Decision Tree Construction for Dataset 2

- Note that we almost forgot Body Temperature attribute.
- How do we deal with such continuous attributes?

Decision Tree Construction for Dataset 2

- Note that we almost forgot Body Temperature attribute.
- How do we deal with such continuous attributes?
- There are multiple ways. We will discuss one possible way.

Decision Tree Construction for Dataset 2

- For the current dataset partitions, we will convert Body Temperature attribute so that we get a simple binary attribute.

PARTITION 1:

Body Temperature (°F)	Visit to Foreign Countries	Antibodies in blood	Disease Presence
104	YES	YES	YES
99	YES	YES	NO
100	NO	YES	YES

PARTITION 2:

Body Temperature (°F)	Visit to Foreign Countries	Antibodies in blood	Disease Presence
100	NO	NO	NO
98	YES	NO	YES
102	YES	NO	NO

Decision Tree Construction for Dataset 2

- For the current dataset partitions, we will convert Body Temperature attribute so that we get a simple binary attribute.

PARTITION 1:

Body Temperature ($>= 100°F$)	Visit to Foreign Countries	Antibodies in blood	Disease Presence
YES	YES	YES	YES
NO	YES	YES	NO
YES	NO	YES	YES

PARTITION 2:

Body Temperature ($>= 100°F$)	Visit to Foreign Countries	Antibodies in blood	Disease Presence
YES	NO	NO	NO
NO	YES	NO	YES
YES	YES	NO	NO

Decision Tree Construction for Dataset 2

PARTITION 1: **PARTITION 1**:

Body Temperature ($>= 100°F$)	Visit to Foreign Countries	Antibodies in blood	Disease Presence
YES	YES	YES	YES
NO	YES	YES	NO
YES	NO	YES	YES

PARTITION 2:

Body Temperature ($>= 100°F$)	Visit to Foreign Countries	Antibodies in blood	Disease Presence
YES	NO	NO	NO
NO	YES	NO	YES
YES	YES	NO	NO

- Note that in each of the partitions, the modified Body Temperature attribute is perfectly correlated with the Disease Presence label.

Decision Tree Construction for Dataset 2

PARTITION 1:

Body Temperature ($>= 100°F$)	Visit to Foreign Countries	Antibodies in blood	Disease Presence
YES	YES	YES	YES
NO	YES	YES	NO
YES	NO	YES	YES

PARTITION 2:

Body Temperature ($>= 100°F$)	Visit to Foreign Countries	Antibodies in blood	Disease Presence
YES	NO	NO	NO
NO	YES	NO	YES
YES	YES	NO	NO

- We can formalize this correlation using the **weighted impurity** for each attribute in PARTITION 1 and PARTITION 2.

Decision Tree Construction for Dataset 2

PARTITION 1:

Body Temperature ($>= 100°F$)	Visit to Foreign Countries	Antibodies in blood	Disease Presence
YES	YES	YES	YES
NO	YES	YES	NO
YES	NO	YES	YES

PARTITION 2:

Body Temperature ($>= 100°F$)	Visit to Foreign Countries	Antibodies in blood	Disease Presence
YES	NO	NO	NO
NO	YES	NO	YES
YES	YES	NO	NO

- **Claim:** In each partition, **weighted impurity** $I(BT)$ of Body Temperature$>= 100°F$ is the lowest.
- **Prove this claim!** (Homework).

Decision Tree Construction for Dataset 2

PARTITION 1:

Body Temperature ($>= 100°F$)	Visit to Foreign Countries	Antibodies in blood	Disease Presence
YES	YES	YES	YES
NO	YES	YES	NO
YES	NO	YES	YES

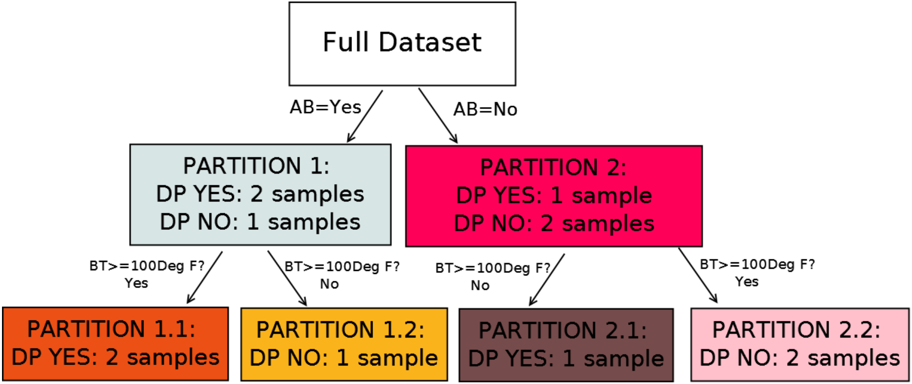
PARTITION 2:

Body Temperature ($>= 100°F$)	Visit to Foreign Countries	Antibodies in blood	Disease Presence
YES	NO	NO	NO
NO	YES	NO	YES
YES	YES	NO	NO

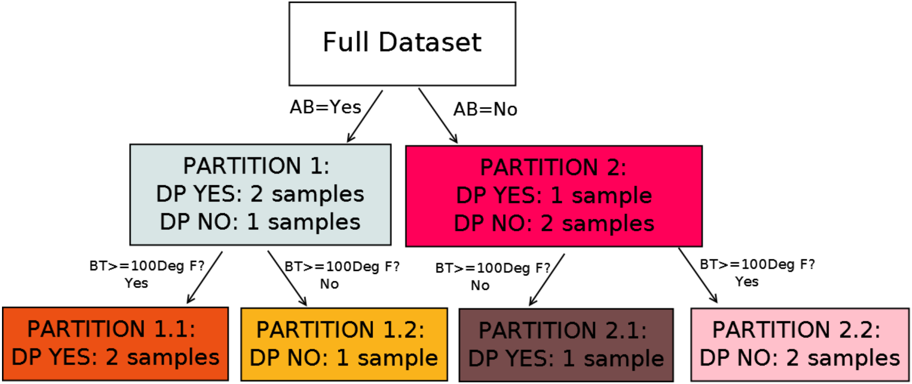
- Thus, each partition will be further split using Body Temperature$>= 100°F$ attribute.

Decision Tree Construction for Dataset 2

- After splitting PARTITION 1 and PARTITION 2 we would get: (**check this!**)



Decision Tree Construction for Dataset 2



- **Note:** PARTITION 1.1, 1.2, 2.1 and 2.2 have samples belonging to only one class.
- There is nothing to split after this in PARTITION 1.1, 1.2, 2.1 and 2.2. Hence we can stop the split procedure.

Decision Tree

Dataset with other types of attributes

Name	Region Visited	Cough Severity	Disease Presence
Nam1	Africa	Low	NO
Nam2	Europe	Medium	YES
Nam3	Europe	High	YES
Nam4	Australia	High	YES
Nam5	Middle-East	Low	NO
Nam6	USA	Low	YES

- Name is a **nominal** attribute.
- Name attribute has distinct value for each sample, hence its utility in classification is very less.
- Attributes having distinct values for each sample will be usually ignored from the splitting procedure (because of their high weighted impurity values).

Decision Tree

Dataset with other types of attributes

Name	Region Visited	Cough Severity	Disease Presence
Nam1	Africa	Low	NO
Nam2	Europe	Medium	YES
Nam3	Europe	High	YES
Nam4	Australia	High	YES
Nam5	Middle-East	Low	NO
Nam6	USA	Low	YES

- Region Visited is a **categorical** attribute since it takes multiple categorical values.
- Cough severity is a **Ordinal** attribute since it takes values that have some ranking (or ordering) associated.

Structure of Decision Tree

- **Root node:** Has no incoming edges and has two or more outgoing edges.
- **Intermediate node:** Has one incoming edge and two or more more outgoing edges.
- **Leaf node:** Has one incoming edge and no outgoing edges.

Note: Each leaf node is associated with a class label.

Decision Tree: A generic algorithm

- Input: Data $D = \{(x^i, y^i)\}_{i=1}^n$, $x^i \in \mathbb{R}^d, y^i \in \{1, 2, \dots, K\}$, $\forall i \in \{1, 2, \dots, n\}$.

Decision Tree: A generic algorithm

- Input: Data $D = \{(x^i, y^i)\}_{i=1}^n$, $x^i \in \mathbb{R}^d, y^i \in \{1, 2, \ldots, K\}$, $\forall i \in \{1, 2, \ldots, n\}$.
- Initialize a node P with full data set D. Assign P as root node of tree.

Decision Tree: A generic algorithm

- Input: Data $D = \{(x^i, y^i)\}_{i=1}^n$, $x^i \in \mathbb{R}^d, y^i \in \{1, 2, \ldots, K\}$, $\forall i \in \{1, 2, \ldots, n\}$.
- Initialize a node P with full data set D. Assign P as root node of tree.
- Create a list $L = [P]$.

Decision Tree: A generic algorithm

- Input: Data $D = \{(x^i, y^i)\}_{i=1}^n$, $x^i \in \mathbb{R}^d, y^i \in \{1, 2, \ldots, K\}$, $\forall i \in \{1, 2, \ldots, n\}$.
- Initialize a node P with full data set D. Assign P as root node of tree.
- Create a list $L = [P]$.
- While list L is not empty do:
  - Extract the first node in list L as Q.

Decision Tree: A generic algorithm

- Input: Data $D = \{(x^i, y^i)\}_{i=1}^n$, $x^i \in \mathbb{R}^d, y^i \in \{1, 2, \ldots, K\}$, $\forall i \in \{1, 2, \ldots, n\}$.
- Initialize a node P with full data set D. Assign P as root node of tree.
- Create a list $L = [P]$.
- While list L is not empty do:
  - Extract the first node in list L as Q.
  - If all samples in node Q have the same label y, label the node Q as y. Q is a leaf node.

Decision Tree: A generic algorithm

- Input: Data $D = \{(x^i, y^i)\}_{i=1}^n$, $x^i \in \mathbb{R}^d, y^i \in \{1, 2, \ldots, K\}$, $\forall i \in \{1, 2, \ldots, n\}$.
- Initialize a node P with full data set D. Assign P as root node of tree.
- Create a list $L = [P]$.
- While list L is not empty do:
  - Extract the first node in list L as Q.
  - If all samples in node Q have the same label y, label the node Q as y. Q is a leaf node.
  - If samples in node Q have different labels, construct a split criterion to split the node Q into child nodes $Q_1, Q_2, \ldots, Q_m$. Add these nodes $Q_1, Q_2, \ldots, Q_m$ at the end of list L.

Decision Tree Construction

What stopping criterion can be used to stop the splitting process?

Decision Tree Construction

What stopping criterion can be used to stop the splitting process?

- Depth based
- Threshold on relative impurity levels of child and parent nodes
- Impurity levels of leaf nodes
- Number of samples in the leaf nodes

Decision Tree Construction

What stopping criterion can be used to stop the splitting process?

- Depth based
- Threshold on relative impurity levels of child and parent nodes
- Impurity levels of leaf nodes
- Number of samples in the leaf nodes

Post-pruning process: Sometimes a full tree is constructed and then subtrees are pruned based on grouping procedures.

Decision Tree

Dataset with other types of attributes

Name	Region Visited	Cough Severity	Disease Presence
Nam1	Africa	Low	NO
Nam2	Europe	Medium	YES
Nam3	Europe	High	YES
Nam4	Australia	High	YES
Nam5	Middle-East	Low	NO
Nam6	USA	Low	YES

- **Homework:** Try to construct a decision tree for this dataset!

Decision Tree in Software

scikit learn | Install | User Guide | API | Examples | Community | More

Prev | Up | Next

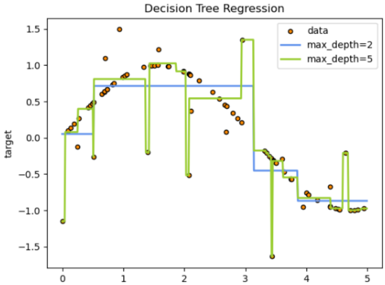
scikit-learn 1.2.0
Other versions

Please **cite us** if you use the software.

1.10. Decision Trees

Decision Trees (DTs) are a non-parametric supervised learning method used for classification an model that predicts the value of a target variable by learning simple decision rules inferred from t seen as a piecewise constant approximation.

For instance, in the example below, decision trees learn from data to approximate a sine curve wi rules. The deeper the tree, the more complex the decision rules and the fitter the model.



Thank You!