

Machine Learning



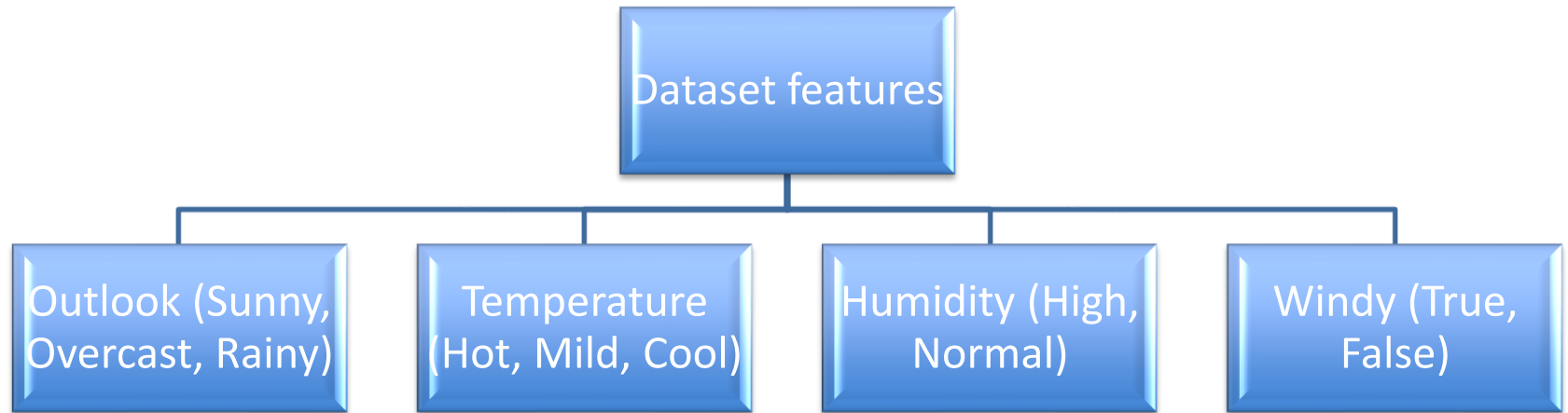
Machine Learning

Lecture: Decision Trees – Worked Example

Ted Scully

Weather Dataset features

- We will use the tennis dataset for illustrating the operation of decision trees
- Weather dataset contains four features that are used to decide whether or not to play tennis



- Objective: Find an hypothesis that describes the cases given and can be used to make decisions in other cases
- Notice that all features above are discrete

Weather Dataset

Anyone for Tennis?					
ID	Outlook	Temp	Humidity	Windy	Play?
A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no

There are 36 ($3 \times 3 \times 2 \times 2$) possible instances with the dataset depicting 14 of those.

Entropy Examples

- In the following slides we will look more closely at the entropy calculations for our weather dataset:
 - **Step 1** – We calculate the **entropy value for the class label of the dataset as a whole**, we will refer to this as the entropy of the weather dataset S.
 - **Step 2** – We calculate the **entropy value of the Windy feature**. We calculate the entropy for Windy = false and for Windy = true

Anyone for Tennis?					
ID	Outlook	Temp	Humidity	Windy	Play?
A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no

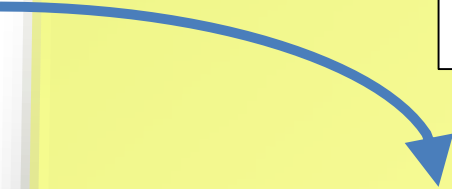
Entropy Step 1 – Entire Dataset

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

- The dataset S refers to our Weather dataset, which has two classes (yes and no).
- We must first determine **the probability of each class i (p_i) in S** .
 - There is a total of **14 examples** in the dataset (the class **yes** occurs **9 times** and the class **no** occurs **5 times**.)
 - Therefore, the probability of class yes is simply **9/14** and the probability of class no is **5/14**. We can now plug these values into the formula and calculate the entropy of the weather dataset.

Entropy Step 1 – Entire Dataset

S refers to
the
Weather
dataset.



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

$$\begin{aligned}\text{Ent}(S) &= -9/14 \log_2(9/14) - \\ &\quad 5/14 \log_2(5/14) \\ &= 0.940\end{aligned}$$

Entropy Step 2 – Windy feature

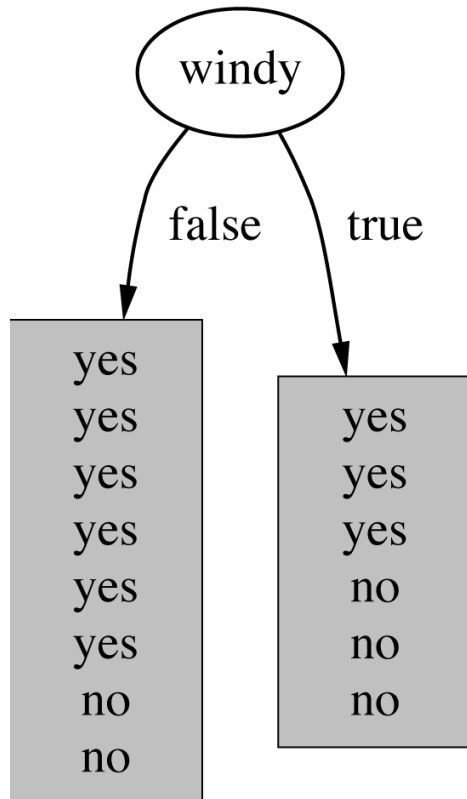
- If we were to divide the data based on the feature Windy. We could need to assess how well this feature splits the data.
- We need to calculate the entropy for the **Windy = true** subset and the **Windy = false** subset.
- In order to calculate the entropy for the **Windy = true** subset we must extract the result (play tennis 'yes' or 'no') when **Windy = true**.
 - This subset is highlighted in green on the next slide.
 - In order to calculate the entropy for the Windy = false subset we must extract the result when Windy = false. This subset is highlighted in yellow in the slide.

Entropy Step 2

Anyone for Tennis?					
ID	Outlook	Temp	Humidity	Windy	Play?
A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no

Entropy Example B

The diagram shows the division of the Weather dataset based on the value of the feature windy.



The grouping on the column on the left shows the output (play tennis 'yes' or 'no') for when **Windy = false** and corresponds to the subset highlighted in green in the previous slide.

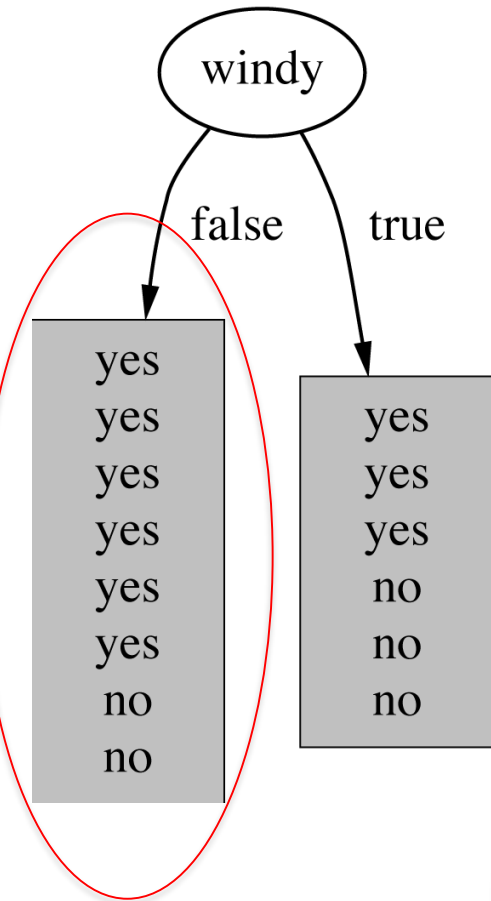
The grouping on the right shows the output for when **Windy = true** and corresponds to the subset highlighted in yellow in the previous slide.

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

Entropy Example B

We now have the information we need to calculate the entropy for **Windy = true** and **Windy = false**.

The **Windy = false** subset we have a total of eight entries, 6 are positive ('yes') and 2 are negative ('no').



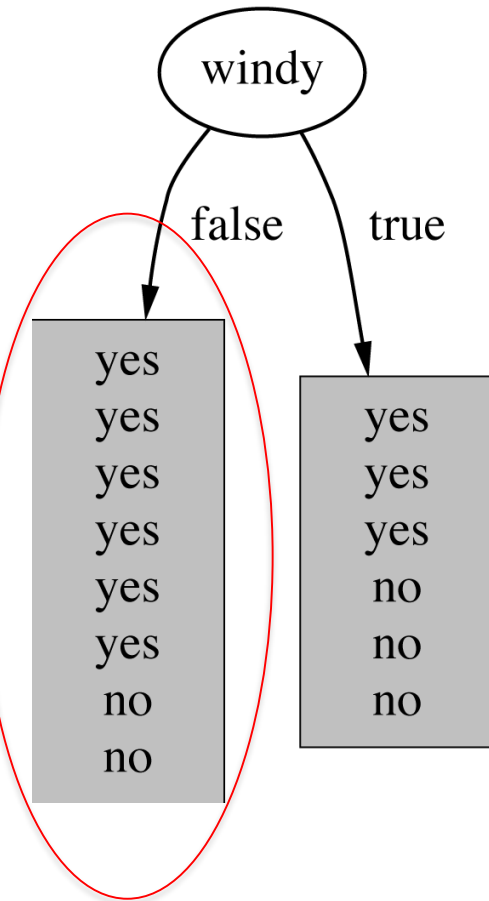
`Ent(S, windy=false)`

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

Entropy Example B

We now have the information we need to calculate the entropy for **Windy = true** and **Windy = false**.

The **Windy = false** subset we have a total of eight entries, 6 are positive ('yes') and 2 are negative ('no').

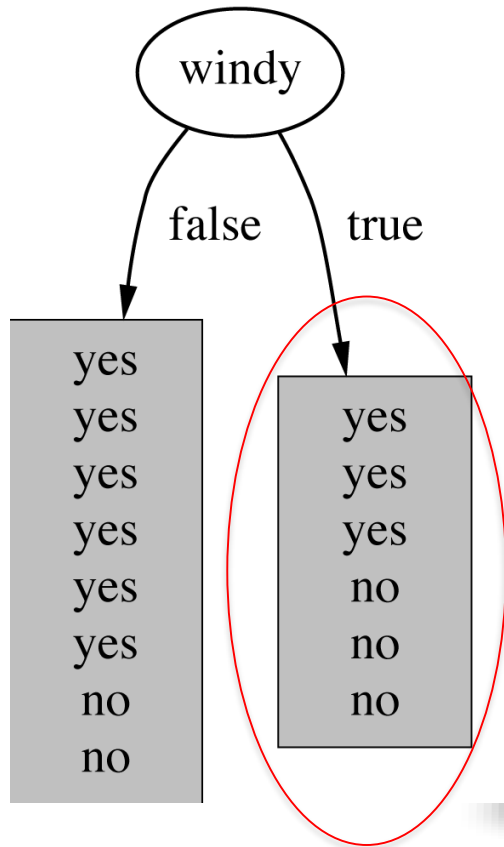


$$\begin{aligned}
 &\text{Ent}(S, \text{windy}=\text{false}) \\
 &= -6/8 \log_2(6/8) - 2/8 \log_2(2/8) \\
 &= 0.3112 + 0.5 = 0.811
 \end{aligned}$$

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

Entropy Example B

The Windy = true subset we have a total of six entries, 3 are positive ('yes') and 3 are negative ('no'). What would you expect the entropy value to be?



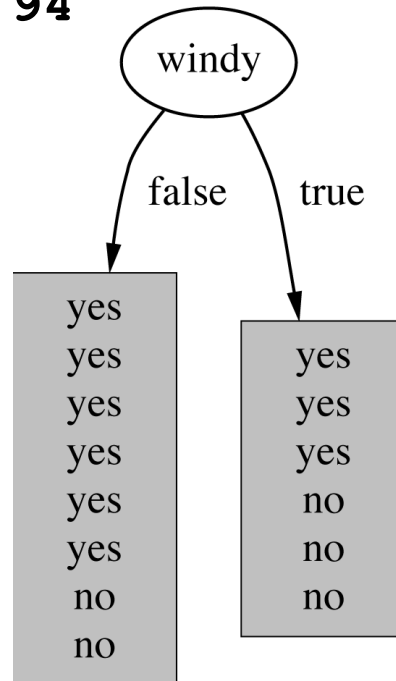
$$\begin{aligned} & \text{Ent}(S, \text{windy}=\text{true}) \\ &= -3/6 \log_2 (3/6) - 3/6 \log_2 (3/6) \\ & \quad 0.5 + 0.5 = 1.0 \end{aligned}$$

Information Gain

- As we mentioned information gain of a feature is the reduction in entropy from dividing the data into two or more subsets based on a particular feature.
- Therefore, in our Weather example the information gain of the feature **Temperature** would be the overall reduction in entropy caused by **partitioning** the data into three subsets: (i) Those where the value is hot; (ii) Those where the value is mild and (iii) Those where the value is cool.
- Ultimately, we are looking for the feature that would lead to the highest information gain (or put another way that would give the largest reduction in entropy).
 - The higher the information gain of a particular feature then the better that feature partitions the data.

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

Ent(S) = 0.94



Ent(S, w=false)=0.811

Ent(S, w=true)= 1.0

Information Gain Example 1

$$\begin{aligned} & \text{InformationGain}(S, \text{Windy}) \\ &= \text{Entropy}(S) - \frac{8}{14} \text{Ent}(S, \text{windy}=\text{false}) \\ & \quad - \frac{6}{14} \text{Ent}(S, \text{windy}=\text{true}) \end{aligned}$$

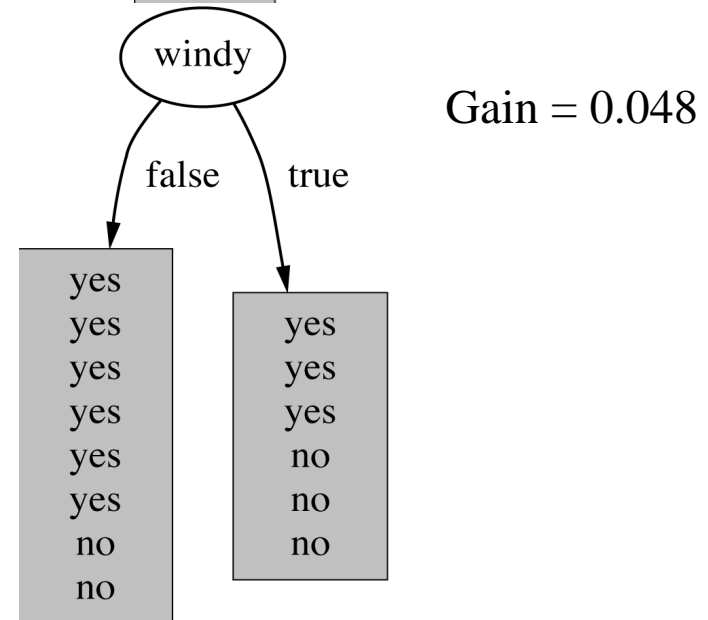
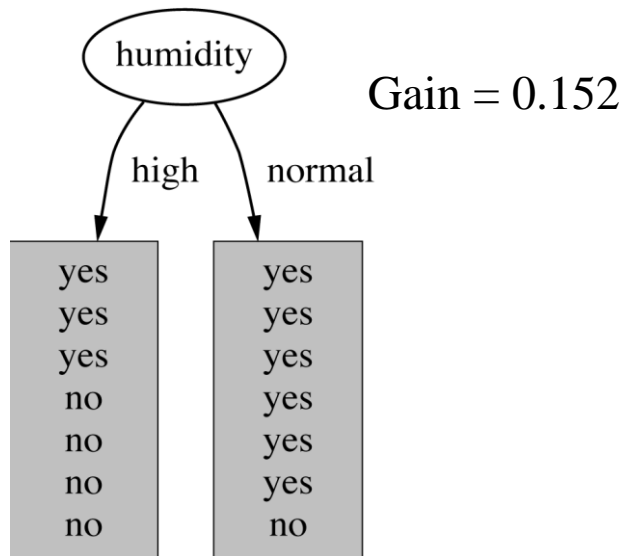
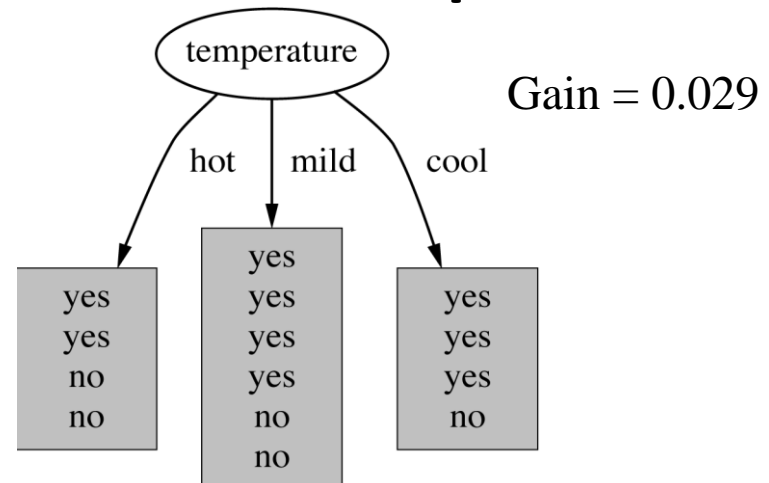
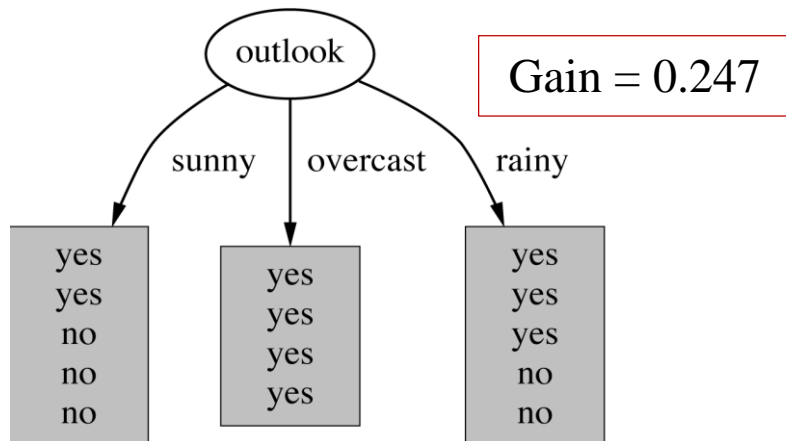
In previous slides we have calculated the entropy value of the weather dataset S as 0.940, the entropy of **Windy = true** as 1.0 and the entropy of **Windy = false** as 0.811. We can now use these values to calculate the information gain for the feature Windy.

Information Gain Example 1

$$\begin{aligned} & \text{InformationGain}(S, \text{Windy}) \\ &= \text{Entropy}(S) - \frac{8}{14} \text{Ent}(S, \text{windy}=\text{false}) \\ & \quad - \frac{6}{14} \text{Ent}(S, \text{windy}=\text{true}) \\ &= 0.940 - \left(\frac{8}{14}\right) 0.811 - \left(\frac{6}{14}\right) 1.00 \\ &= 0.048 \end{aligned}$$

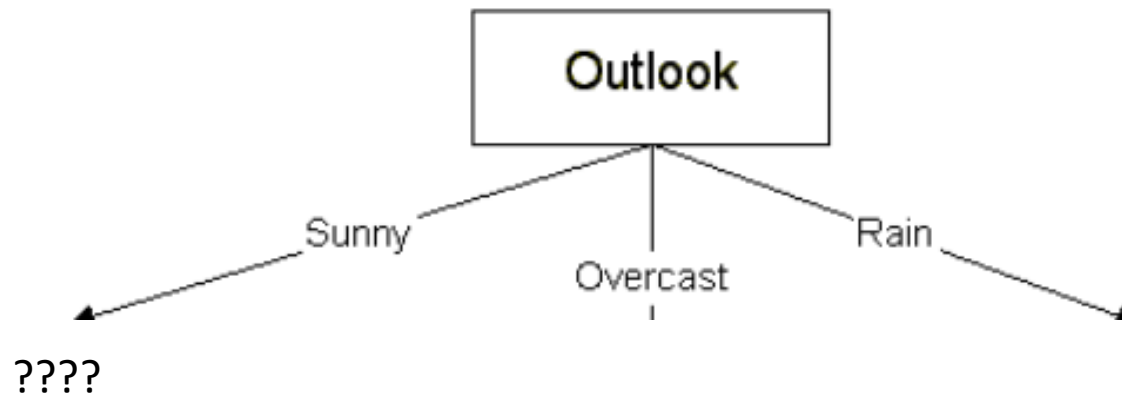
The information gain for Windy is 0.048, which is quite low and means that the Windy feature does not partition the data very well. (The higher the information gain value the better)

Information Gain Example 2



Adding Nodes to the Decision Tree

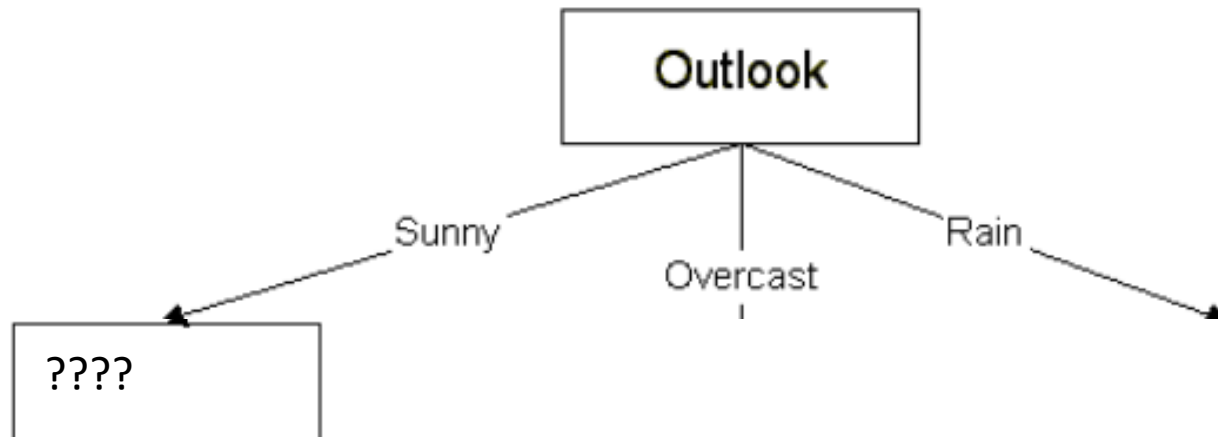
- The previous slides shows the information gain for each feature in the weather dataset.
- The next step is to “select the feature that has the highest information gain” and make it the root node of our decision tree.
- The **outlook feature has the highest information gain** value and therefore becomes a root node.
- We split our tree based on the three possible values of the outlook feature



Adding Nodes to the Decision Tree

- Lets find the best feature to place on the Sunny branch of our decision tree below.
- To do this we will need to calculate the information gain for all remaining features (**Humidity**, **Temperature**, **Windy**) for the data where Outlook = Sunny

ID	Outlook	Temp	Humidity	Windy	Play?
A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no



Information Gain for Humidity feature

- Consider the Gain(Sunny, Humidity). Any observations?

ID	Outlook	Humidity	Play?
A	sunny	high	no
B	sunny	high	no
C	overcast	high	yes
D	rainy	high	yes
E	rainy	normal	yes
F	rainy	normal	no
G	overcast	normal	yes
H	sunny	high	no
I	sunny	normal	yes
J	rainy	normal	yes
K	sunny	normal	yes
L	overcast	high	yes
M	overcast	normal	yes
N	rainy	high	no

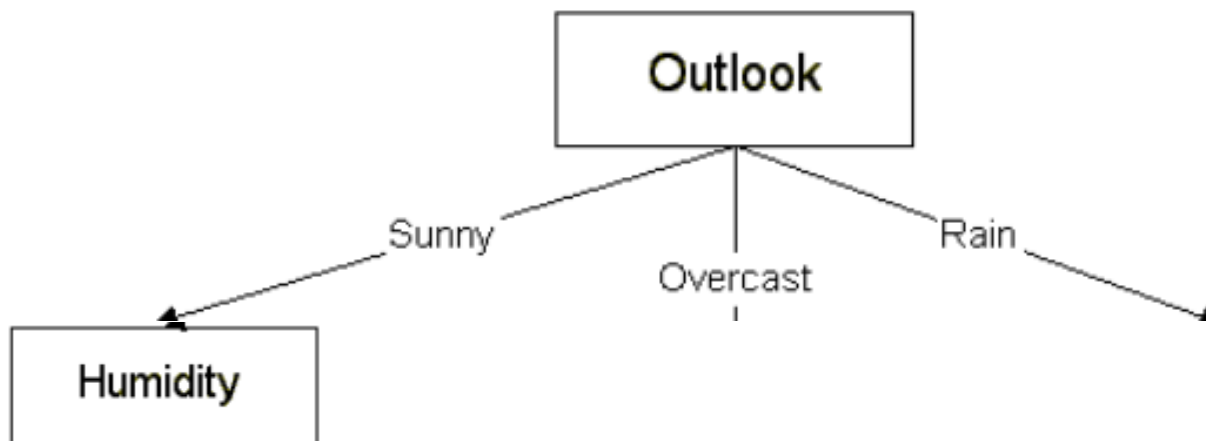
Information Gain for Humidity feature

- $\text{Ent}(\text{Sunny}) = -((2/5) * \log_2(2/5)) - ((3/5) * \log_2(3/5)) = 0.97095059445$
 - (5 Data items where Sunny = yes and no (3 where it is no and 2 where it is yes))
- $\text{Ent}(\text{Sunny}, \text{Humidity} = \text{High}) = 0$
 - (3 data items where sunny and humidity = high, of those 3 they all had an outcome of play = no)
- $\text{Ent}(\text{Sunny}, \text{Humidity} = \text{Normal}) = 0$
 - (2 data items where sunny and humidity = normal, of those 2 they all have an outcome of play = yes)
- $\text{Gain}(\text{Sunny}, \text{Humidity}) = 0.97095059445 - ((3/5)0 + (2/5)0) = 0.97095059445$

ID	Outlook	Humidity	Play?
A	sunny	high	no
B	sunny	high	no
C	overcast	high	yes
D	rainy	high	yes
E	rainy	normal	yes
F	rainy	normal	no
G	overcast	normal	yes
H	sunny	high	no
I	sunny	normal	yes
J	rainy	normal	yes
K	sunny	normal	yes
L	overcast	high	yes
M	overcast	normal	yes
N	rainy	high	no

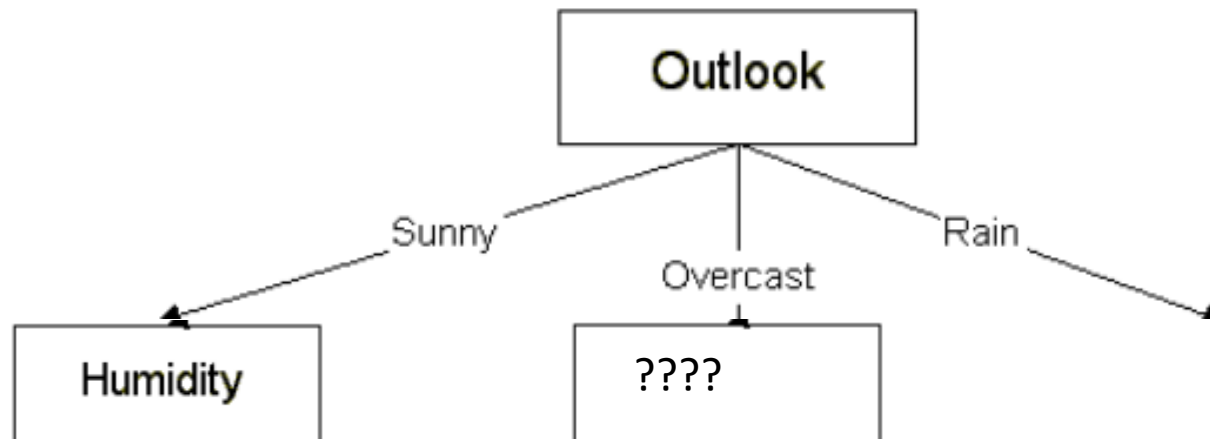
Adding Nodes to the Decision Tree

- Lets now look at applying ID3 to the subset of the dataset where Outlook is Sunny
 - There are 5 examples from table 1 with outlook = sunny
 - $\text{Gain}(\text{Sunny}, \text{Humidity}) = 0.970$
 - $\text{Gain}(\text{Sunny}, \text{Temperature}) = 0.570$
 - $\text{Gain}(\text{Sunny}, \text{Wind}) = 0.019$
- What should be the new node added to the decision tree



Adding Nodes to the Decision Tree

- Lets find the best feature to place on the **Overcast** branch of our decision tree below.



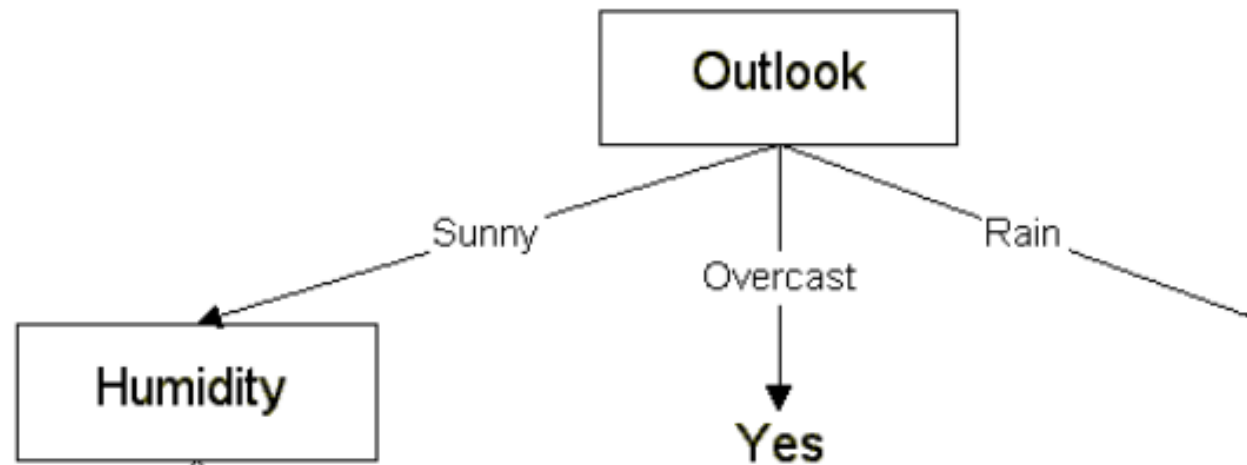
Adding Nodes to the Decision Tree

- Any observations on the subset of the dataset where outlook is overcast?

B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no

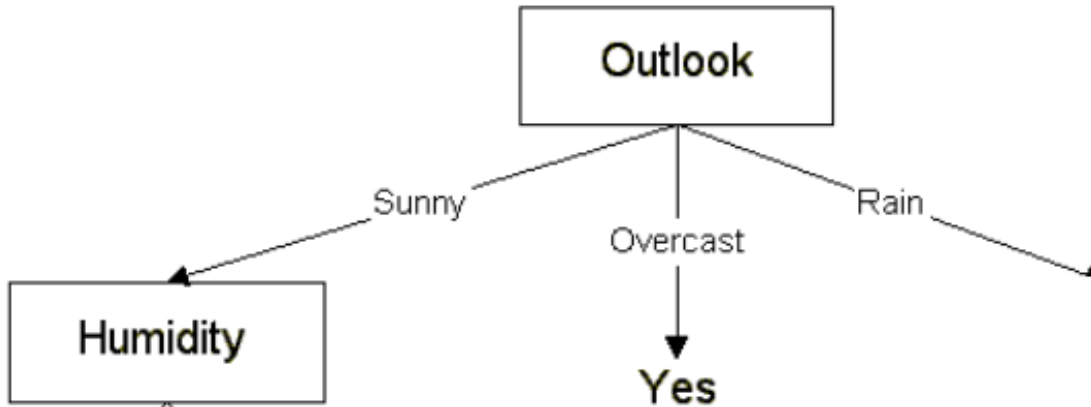
When we come to apply ID3 to the subset where Outlook=overcast we will find that further partitioning of this subset is not needed, as all the cases have the same class

Adding Nodes to the Decision Tree



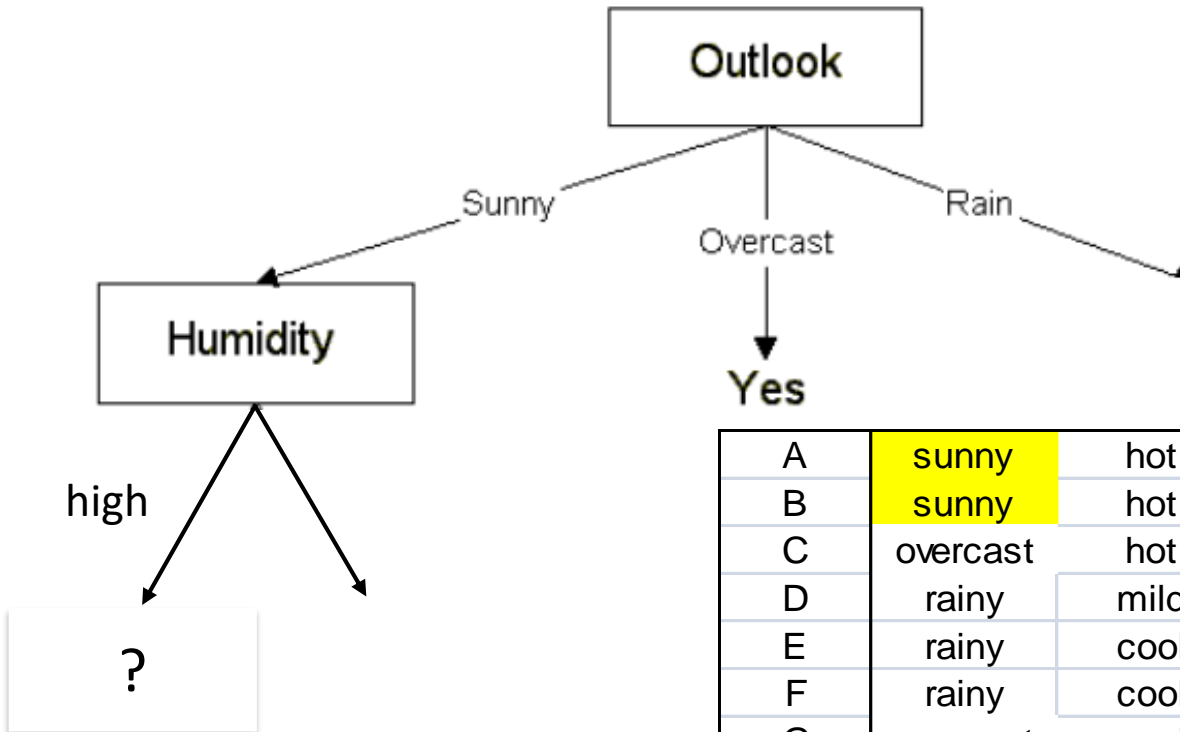
Exercise

- Determine the structure of our decision tree **after the humidity node**.



A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no

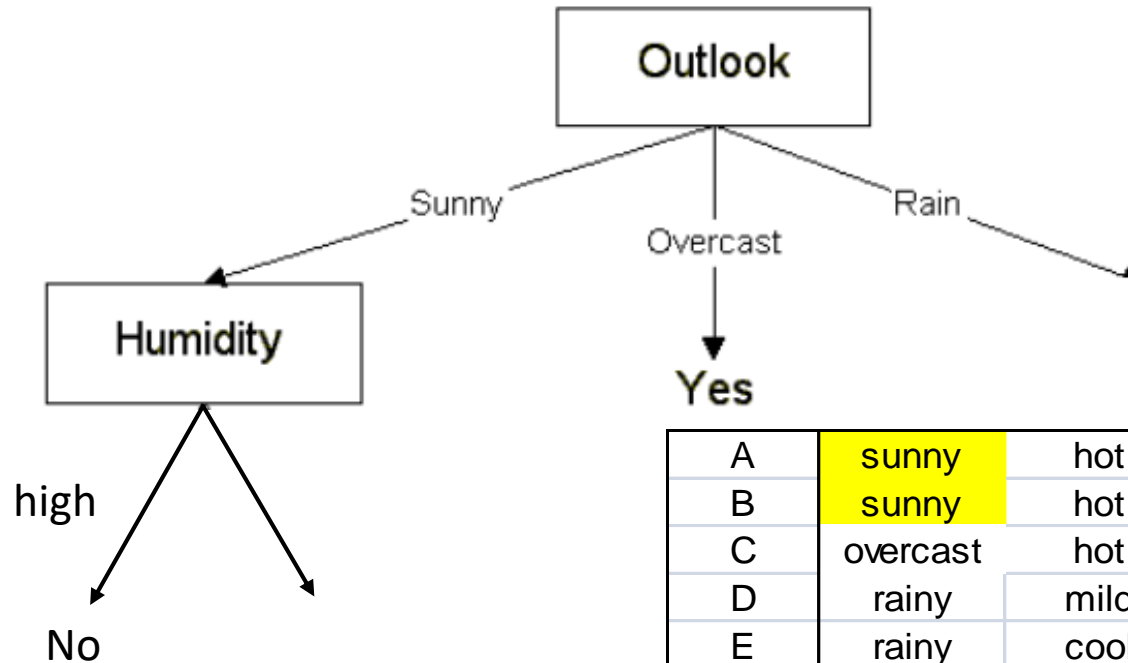
Exercise



- Determine the structure of our decision tree after the humidity node.

A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no

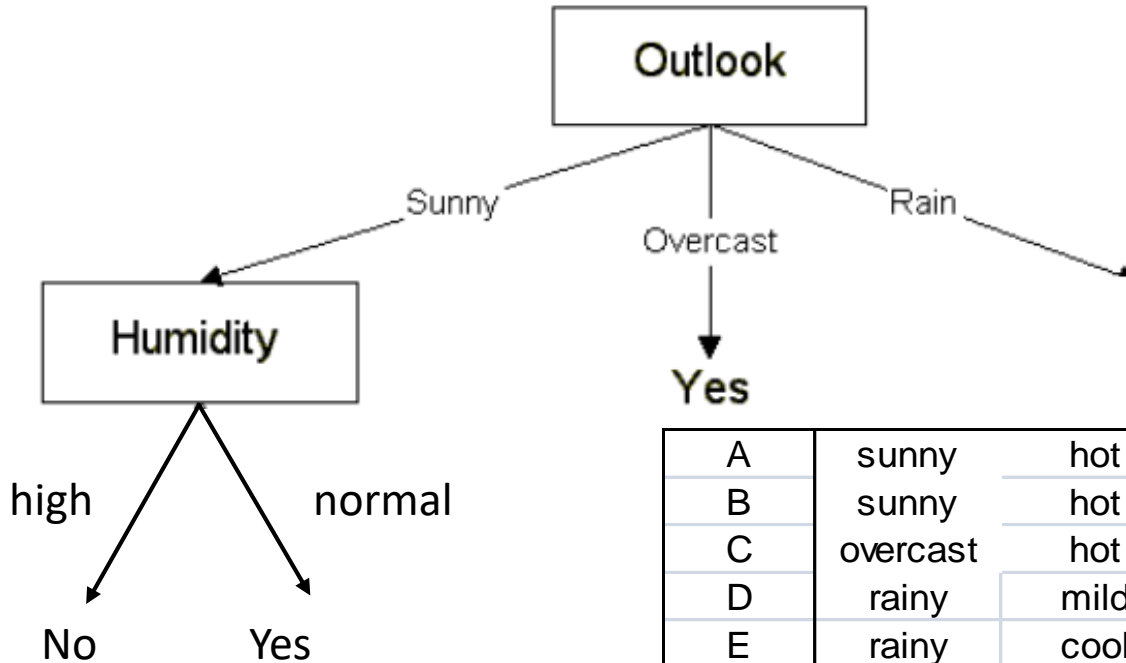
Exercise



- Determine the structure of our decision tree after the humidity node.

A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no

Exercise



- Determine the structure of our decision tree after the humidity node.

A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no

ID3 Algorithm

ID3(Examples, features, Target):

Input:	Examples:	set of classified examples
	features:	set of features in the examples
	Target:	classification to be predicted

if all Examples have same class **then return** this class

else if all features are tested **then return** majority class

else:

let Best = choosefeature()

 #selects feature that **best separates** Examples relative to Target

let Tree = new decision tree with Best as root node

foreach value v_i **of** Best

let Examples _{i} = subset of Examples that have Best= v_i

let Subtree = **ID3(Examples _{i} , features-Best, Target)**

add branch from Tree to Subtree with label v_i

return Tree

Different Metrics for Splitting Data

- We have looked at using information gain and entropy formulas as a means of identify the best feature to split the data.
 - For a dataset S with n different classes(outputs), probability (relative frequency) of class i is denoted p_i

- Gini Index:
$$GI(t) = 1 - \sum_{i=1}^n [p_i]^2$$

The Gini index can be understood as calculating **how often the target in a dataset would be misclassified if predictions were made based on the distribution of the target in the dataset**. For example, if there were two classes that occurred in the dataset with equal frequency then the expected rate of misclassification would be 0.5. Again the objective here is to minimise the Gini Index. If all instance in a dataset belong to one class the Gini Index would be 0.

$$GI(t) = 1 - \sum_{i=1}^n [p_i]^2$$

The Gini index can be understood as calculating **how often the target in a dataset would be misclassified if predictions were made based on the distribution of the target in the dataset.**

Node N_1	Count
Class=0	0
Class=1	6

Node N_2	Count
Class=0	1
Class=1	5

Node N_3	Count
Class=0	3
Class=1	3

$$GI(t) = 1 - \sum_{i=1}^n [p_i]^2$$

The Gini index can be understood as calculating **how often the target in a dataset would be misclassified if predictions were made based on the distribution of the target in the dataset.**

Node N_1	Count
Class=0	0
Class=1	6

$$\text{Gini} = 1 - (0/6)^2 - (6/6)^2 = 0$$

Node N_2	Count
Class=0	1
Class=1	5

$$\text{Gini} = 1 - (1/6)^2 - (5/6)^2 = 0.278$$

Node N_3	Count
Class=0	3
Class=1	3

$$\text{Gini} = 1 - (3/6)^2 - (3/6)^2 = 0.5$$

Information Gain

- Note: The Gini Index and the Misclassification Error metrics can be plugged into the Information Gain function in the same way as the entropy metric.

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

Subset of S
where A has
value v

Where the function M can be either the **Entropy** function, the **Gini** Index or the **Misclassification Error**

Summary

- A decision tree is an easily interpretable machine learning model (we can represent its decision process graphically in tree form).
- The process of training a decision tree is focus on selecting the “best” feature to use a node.
- Typically, we calculate the information gain. That is we look at the difference in uncertainty in the data before we split us and after we split using a specific feature.
- Commonly used metrics for splitting data:
 - Entropy
 - Gini
 - Misclassification Error
- There are many different algorithms for building decision trees such as ID3, CART and they have been used very effectively in bagging and boosting ensembles.