machine learning algorithms are based on the tree structure

- starts at root and branches off
- flowchart-like operation
  - starts a the root node with a specific question about the data
  - each branch leads to a potential answer/approach
  - the branches then lead to decision (internal nodes) which ask more questions to narrow it down

  “Pasted image 20240205140234.png” could not be found.

this is an effective method for making decisions

- lay out the problem and all possible outcomes

## Trees terminology in ml

- root - represents entire message or decision
- decision (internal) node - a node where the prior node branched into one of two variables
- leaf (terminal) node -
- splitting - the process of dividing a node into two or more nodes
- pruning - the opposite of splitting, the process of going through and reducing the tree to only the most important nodes or outcomes

# Classification Trees

tree to classify -> arrive at a specific output

# Regression Trees

algorithms to predict what is likely to happen given some information

- example: housing prices in Colorado
  - input - what prices have been in previous years
  - output - house prices in the next year

# Construction of a Decision Tree

1. select an attribute to place at the root node and make one branch for each possible value

2. split up the example set into subsets, one for each
   ...

# Examples

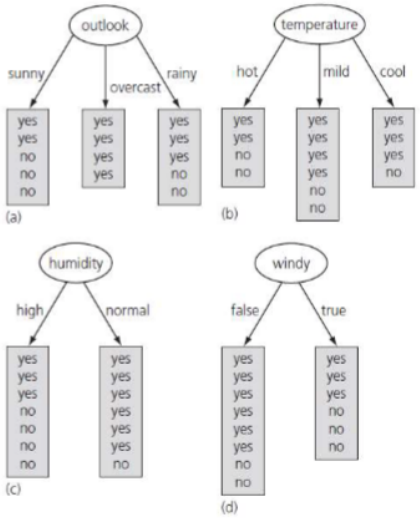
“Pasted image 20240205141153.png” could not be found.

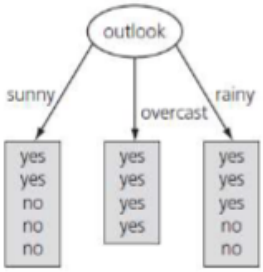
## how to determine which attribute to split on

- any leaf with only one class will not have to be split futrther
- a measure of purity of each node may be useful
  - information in the unit of bits
  - the expected amount of information that would be needed to describe information

- **How to determine which attribute to split on**
  - The weather data for decision tree

| Outlook | Temperature | Humidity | Windy | Play |
|---|---|---|---|---|
| sunny | hot | high | false | no |
| sunny | hot | high | true | no |
| overcast | hot | high | false | yes |
| rainy | mild | high | false | yes |
| rainy | cool | normal | false | yes |
| rainy | cool | normal | true | no |
| overcast | cool | normal | true | yes |
| sunny | mild | high | false | no |
| sunny | cool | normal | false | yes |
| rainy | mild | normal | false | yes |
| sunny | mild | normal | true | yes |
| overcast | mild | high | true | yes |
| overcast | hot | normal | false | yes |
| rainy | mild | high | true | no |




-The numbers of yes and no classes at the leaf nodes are [2,3], [4,0], and [3,2]
-The information values of these nodes
ent([2, 3]) = 0.971 bits = -(2/5)*log(2/5)-(3/5)*log(3/5)
ent([4, 0]) = 0.0 bits
ent([3, 2]) = 0.971 bits

Calculating Uncertainty

- Requested Properties of uncertainty
  - When the number of either yes's or no's is zero, the uncertainty is zero.
  - When the number of yes's and no's is equal, the uncertainty reaches a maximum.
  - The uncertainty must obey the multistage property
- Uncertainty (**entropy)**
  - **Degree of uncertainty**

$$\text{entropy}(p_1, p_2, \ldots, p_n) = -p_1 \log p_1 - p_2 \log p_2 \ldots - p_n \log p_n$$

$$\text{entropy}(p,q,r) = \text{entropy}(p,q+r) + (q+r) \cdot \text{entropy}\left(\frac{q}{q+r}, \frac{r}{q+r}\right)$$

# Limitation of information gain

accroding to this rule, we ought to select the root node with the most children (most information gained from a node). However, some attributes are not relavent to our particular decision and we may may go down a root node with many children that doesn't help us at all with our problem

# The Gain Ratio

tries to counter the information gain problem by taking into account for the issues with information gain

- taking into account the number and size of daughter nodes
- calculated by dividing the original information gain by the uncertainty of the attribute

## Issue with information gain

- **Limitation of the gain ratio**
  - The gain ratio can lead to preferring an attribute just because its intrinsic information is much lower than that for the other attributes
  - A standard fix
    - **Choose the attribute that maximizes the gain ratio**, provided that the **information gain for that attribute is at least as great as the average information gain for all the attributes examined**
- **C4.5**
  - A series of improvements to information gain
  - **Including methods for dealing with numeric attributes, missing values, noisy data, and generating rules from trees**

# ID3

a top down greedy search algorithm. We replace information gain with a **standard deviation reduction** strategy. The idea is to partition the data into subsets that contain instances with similar values (homogeneous).

- standard deviation (s) -> branching
- Coefficient of Deviation (CV) or Count (n) -> stop branching

## Standard deviation for one attribute

- ID3
  - Standard deviation for one attribute

| Hours Played |
|---|
| 25 |
| 30 |
| 46 |
| 45 |
| 52 |
| 23 |
| 43 |
| 35 |
| 38 |
| 46 |
| 48 |
| 52 |
| 44 |
| 30 |

$$Count = n = 14$$

$$Average = \bar{x} = \frac{\sum x}{n} = 39.8$$

$$Standard\ Deviation = S = \sqrt{\frac{\sum (x - \bar{x})^2}{n}} = 9.32$$

$$Coeffeicient\ of\ Variation = CV = \frac{S}{\bar{x}} * 100\% = 23\%$$

$$CV = \frac{Standard\ Devication}{Average}$$

https://saedsayad.com/decision_tree_reg.htm

- we do this to scale the so one homogenous group doesn't dominate another based on the magnitude of the values of that happen to be in a particular group

## Calculating Std. Deviation for 2 attributes

$$S(T, X) = \sum P(c)S(c)$$

| | | Hours Played (StDev) | Count |
|---|---|---|---|
| Outlook | Overcast | 3.49 | 4 |
| | Rainy | 7.78 | 5 |
| | Sunny | 10.87 | 5 |
| | | | 14 |



**S**(Hours, Outlook) = **P**(Sunny)***S**(Sunny) + **P**(Overcast)***S**(Overcast) + **P**(Rainy)***S**(Rainy)

= (4/14)*3.49 + (5/14)*7.78 + (5/14)*10.87

= 7.66

https://saedsayad.com/decision_tree_reg.htm

## Std. Deviation Reduction Technique

based on the **decrease in standard deviation after a dataset is split** based on an attribute.

- Step 1: **The standard deviation of the target** is calculated
  - ➢ Standard deviation (Hours Played) = 9.32

Step 2: **Split the dataset** on the different attributes and **calculate the standard deviation** for each branch, and **finally calculate the standard deviation reduction** by subtracting the resulting standard deviation from the standard deviation before the split.

| | | Hours Played (StDev) |
|---|---|---|
| Outlook | Overcast | 3.49 |
| | Rainy | 7.78 |
| | Sunny | 10.87 |
| SDR=1.66 | | |

| | | Hours Played (StDev) |
|---|---|---|
| Temp. | Cool | 10.51 |
| | Hot | 8.95 |
| | Mild | 7.65 |
| SDR= **0.48** | | |

| | | Hours Played (StDev) |
|---|---|---|
| Humidity | High | 9.36 |
| | Normal | 8.37 |
| SDR=0.28 | | |

| | | Hours Played (StDev) |
|---|---|---|
| Windy | False | 7.87 |
| | True | 10.59 |
| SDR=0.29 | | |

$$SDR(T,X) = S(T) - S(T,X)$$

**SDR**(Hours , Outlook) = **S**(Hours ) – **S**(Hours, Outlook)

= 9.32 – 7.66 = 1.66

https://saedsayad.com/decision_tree_reg.htm

Step 3: Choose **the attribute with the largest standard deviation reduction for the decision node**

| ★ | | Hours Played (StDev) |
|---|---|---|
| Outlook | Overcast | 3.49 |
| | Rainy | 7.78 |
| | Sunny | 10.87 |
| SDR=1.66 | | |

https://saedsayad.com/decision_tree_reg.htm

Step 4a: **Divide the dataset based on the values of the selected attribute**.

- This process is run recursively on the non-leaf branches, until all data is processed.
- **Some termination criteria is required not to have very few instances**


| Outlook | Temp | Humidity | Windy | Hours Played |
|---|---|---|---|---|
| Sunny | Mild | High | FALSE | 45 |
| Sunny | Cool | Normal | FALSE | 52 |
| Sunny | Cool | Normal | TRUE | 23 |
| Sunny | Mild | Normal | FALSE | 46 |
| Sunny | Mild | High | TRUE | 30 |
| Overcast | Hot | High | FALSE | 46 |
| Overcast | Cool | Normal | TRUE | 43 |
| Overcast | Mild | High | TRUE | 52 |
| Overcast | Hot | Normal | FALSE | 44 |
| Rainy | Hot | High | FALSE | 25 |
| Rainy | Hot | High | TRUE | 30 |
| Rainy | Mild | High | FALSE | 35 |
| Rainy | Cool | Normal | FALSE | 38 |
| Rainy | Mild | Normal | TRUE | 48 |

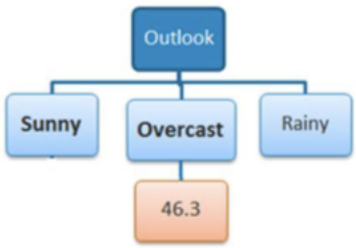
https://saedsayad.com/decision_tree_reg.htm

Step 4b: **"Overcast" subset** does not need any further splitting because its **CV (8%) is less than the threshold (10%) → set it as leaf and determine its value**

- The related leaf node gets the average of the "Overcast" subset.

Outlook - Overcast

| | | Hours Played (StDev) | Hours Played (AVG) | Hours Played (CV) | Count |
|---|---|---|---|---|---|
| Outlook | Overcast | 3.49 | 46.3 | 8% | 4 |
| | Rainy | 7.78 | 35.2 | 22% | 5 |
| | Sunny | 10.87 | 39.2 | 28% | 5 |




https://saedsayad.com/decision_tree_reg.htm

Step 4c: However, the "Sunny" branch has an CV (28%) more than the threshold (10%) which needs further splitting.

- We select "Windy" as the best node after "Outlook" because it has the largest SDR.

Outlook - Sunny

| Temp | Humidity | Windy | Hours Played |
|---|---|---|---|
| Mild | High | FALSE | 45 |
| Cool | Normal | FALSE | 52 |
| Cool | Normal | TRUE | 23 |
| Mild | Normal | FALSE | 46 |
| Mild | High | TRUE | 30 |
| | | | S = 10.87 |
| | | | AVG = 39.2 |
| | | | CV = 28% |

| | | Hours Played (StDev) | Count |
|---|---|---|---|
| Temp | Cool | 14.50 | 2 |
| | Mild | 7.32 | 3 |

SDR = 10.87-((2/5)*14.5 + (3/5)*7.32) = **0.678**

| | | Hours Played (StDev) | Count |
|---|---|---|---|
| Humidity | High | 7.50 | 2 |
| | Normal | 12.50 | 3 |

SDR = 10.87-((2/5)*7.5 + (3/5)*12.5) = **0.370**

| | | Hours Played (StDev) | Count |
|---|---|---|---|
| Windy | False | 3.09 | 3 |
| | True | 3.50 | 2 |

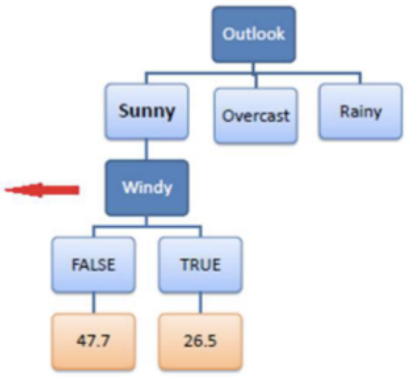
SDR = 10.87-((3/5)*3.09 + (2/5)*3.5) = **7.62**

https://saedsayad.com/decision_tree_reg.htm

Step 4c: However, the "Sunny" branch has an CV (28%) more than the threshold (10%) which needs further splitting.

- Because **the number of data points** for both branches (FALSE and TRUE) in Windy is **equal o less than 3 we stop** further branching and assign the average of each branch to the related leaf node.

| Temp | Humidity | Windy | Hours Played |
|---|---|---|---|
| Mild | High | FALSE | 45 |
| Cool | Normal | FALSE | 52 |
| Mild | Normal | FALSE | 46 |
| Cool | Normal | TRUE | 23 |
| Mild | High | TRUE | 30 |




https://saedsayad.com/decision_tree_reg.htm

- Step 4d: Moreover, **the "rainy" branch** has an CV (22%) which is more than the threshold (10%). This branch needs further splitting. We select **"Temp" as the best node**

Outlook - Rainy

| Temp | Humidity | Windy | Hours Played |
|---|---|---|---|
| Hot | High | FALSE | **25** |
| Hot | High | TRUE | **30** |
| Mild | High | FALSE | **35** |
| Cool | Normal | FALSE | **38** |
| Mild | Normal | TRUE | **48** |
| | | | **S** = 7.78 |
| | | | **AVG** = 35.2 |
| | | | **CV** = 22% |

| | | Hours Played (StDev) | Count |
|---|---|---|---|
| Temp | Cool | 0 | 1 |
| | Hot | 2.5 | 2 |
| | Mild | 6.5 | 2 |

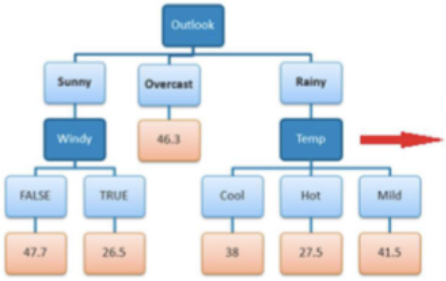
SDR = 7.78 - [(1/5)*0+(2/5)*2.5 + (2/5)*6.5] = **4.18**

| | | Hours Played (StDev) | Count |
|---|---|---|---|
| Humidity | High | 4.1 | 3 |
| | Normal | 5.0 | 2 |

SDR = 7.78 - [(3/5)*4.1 + (2/5)*5.0] = **3.32**

| | | Hours Played (StDev) | Count |
|---|---|---|---|
| Windy | False | 5.6 | 3 |
| | True | 9.0 | 2 |

SDR = 7.78 - [(3/5)*5.6 + (2/5)*9.0] = **0.82**




| Temp | Hours Played |
|---|---|
| Cool | 38 |
| Hot | 25 |
| Hot | 30 |
| Mild | 35 |
| Mild | 48 |

https://saedsayad.com/decision_tree_reg.htm