CSE- 443

Pattern Recognition

Ref. Book:

- Pattern Classification (2nd Edition) -R. O. Duda, P.E.D. Hart and G. Stork; John Wiley and Sons (2000)
- Pattern recognition (4th Edition) –Sergios Theodoridis and Konstantinos Koutroumbas; Academic Press (2008)

Classification

Basic Concepts

Decision Tree Classification

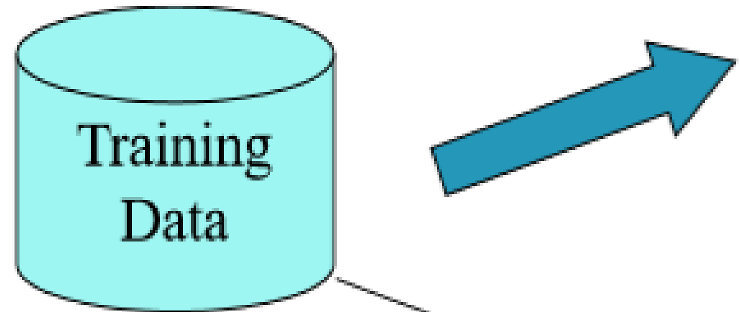
Supervised vs. Unsupervised Learning

- Supervised learning (classification)
  - Supervision: The training data (observations, measurements, etc.) are accompanied by **labels** indicating the class of the observations
  - New data is classified based on the training set
- Unsupervised learning (clustering)
  - The class labels of training data is unknown
  - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

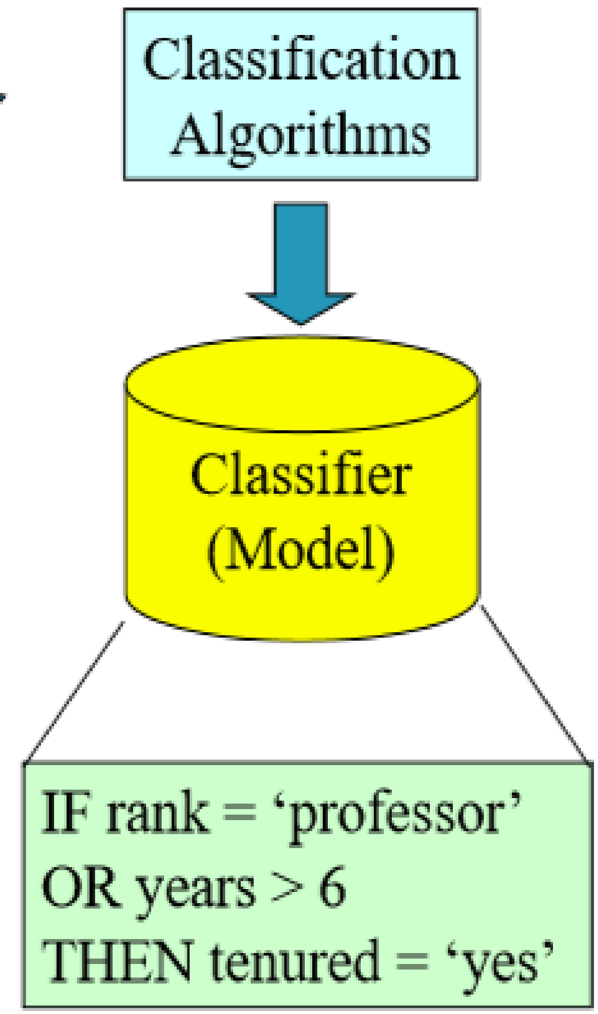
Classification—A Two-Step Process

- Model construction: describing a set of predetermined classes
  - Each tuple/sample is assumed to belong to a predefined class, as determined by the class label attribute
  - The set of tuples used for model construction is training set
  - The model is represented as classification rules, decision trees, or mathematical formulae
- Model usage: for classifying future or unknown objects
  - Estimate accuracy of the model
    - The known label of test sample is compared with the classified result from the model
    - Accuracy rate is the percentage of test set samples that are correctly classified by the model
    - Test set is independent of training set (otherwise overfitting)
  - If the accuracy is acceptable, use the model to classify new data
- Note: If *the test set* is used to select models, it is called validation (test) set

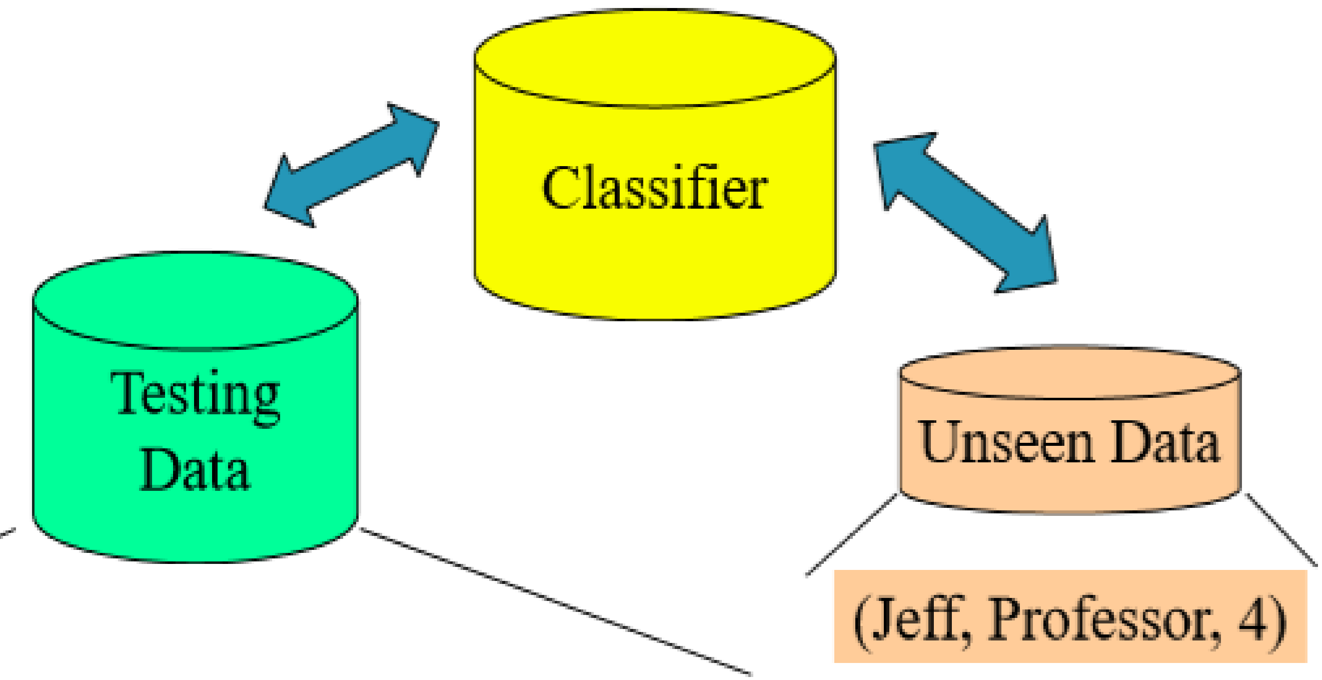
Process (1): Model Construction



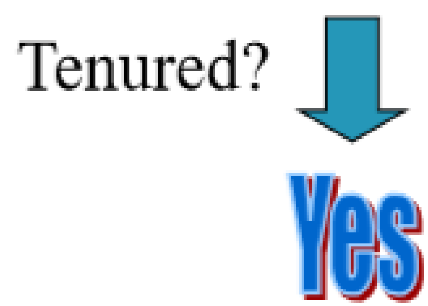
NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no



Process (2): Using the Model in Prediction



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes



Decision Tree

- Decision tree is a versatile classifier and could be used in many applications.

- The leaves in classical decision tress are the nodes associated with labels,

  i.e.: predict decision, while the internal nodes are feature nodes with split the data into its children.

- The main concern is to construct a decision tree with minimum height or size, given the training data.

Decision Tree

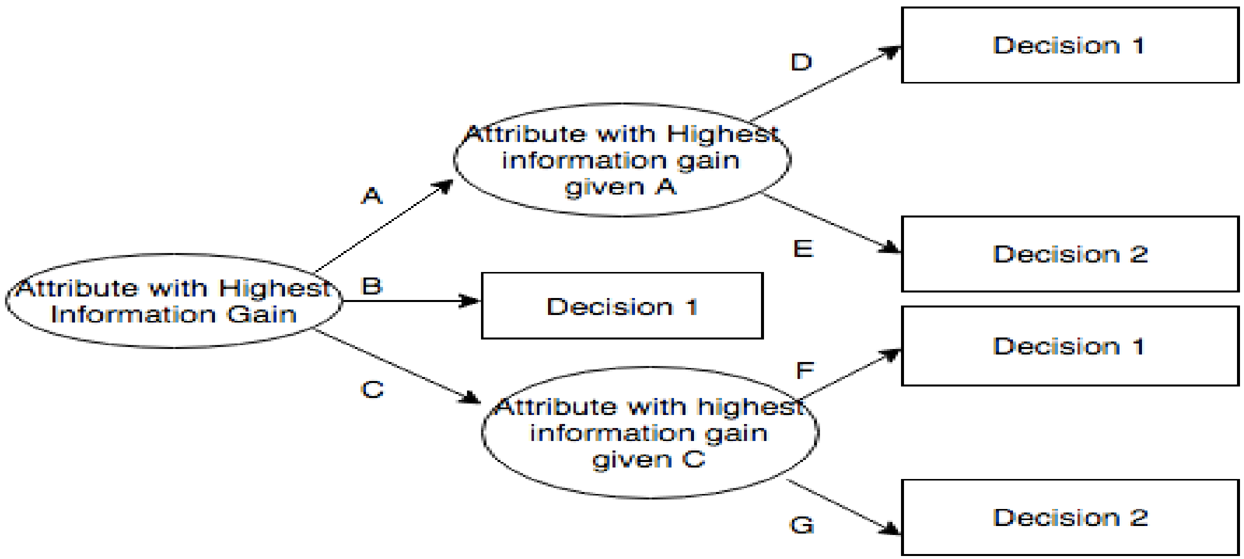
- Decision tree builds regression or classification models in the form of a tree structure.
- It breaks down a dataset into smaller subsets while at the same time an associated decision tree is incrementally developed.
- The final result is a tree with **decision nodes** and **leaf nodes**. The topmost decision node in a tree which corresponds to the best predictor called **root node**.
- Decision trees can handle both categorical and numerical data.

Regression:

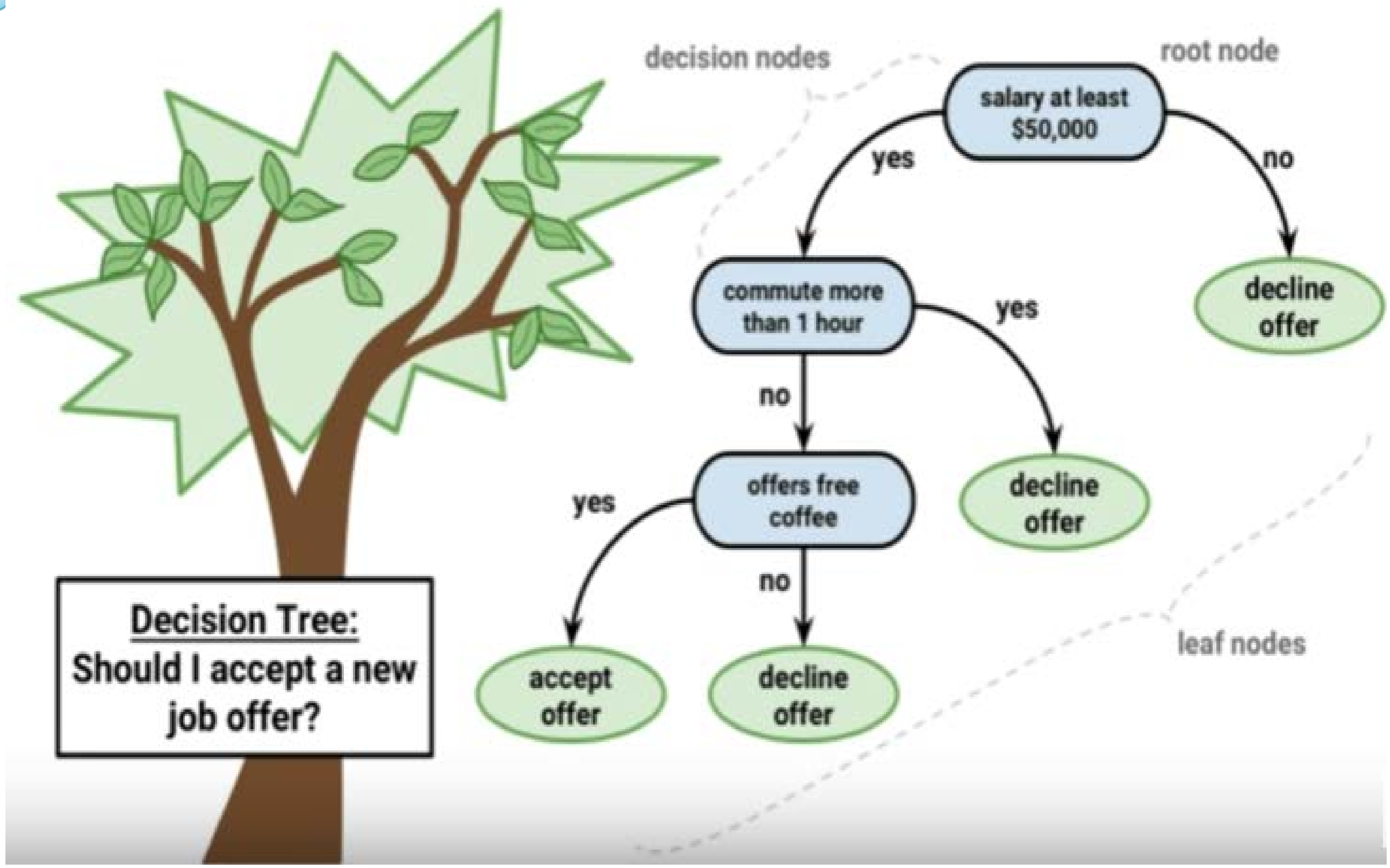
A measure of the relation between the mean value of one variable (e.g. output) and corresponding values of other variables (e.g. time and cost).

Decision Tree

- The core algorithm for building decision trees called **ID3** (Iterative Dichotomiser 3**)** by J. R. Quinlan which employs a top-down, greedy search through the space of possible branches with no backtracking.
- The ID3 algorithm can be used to construct a decision tree for regression by replacing Information Gain with *Standard Deviation Reduction*.

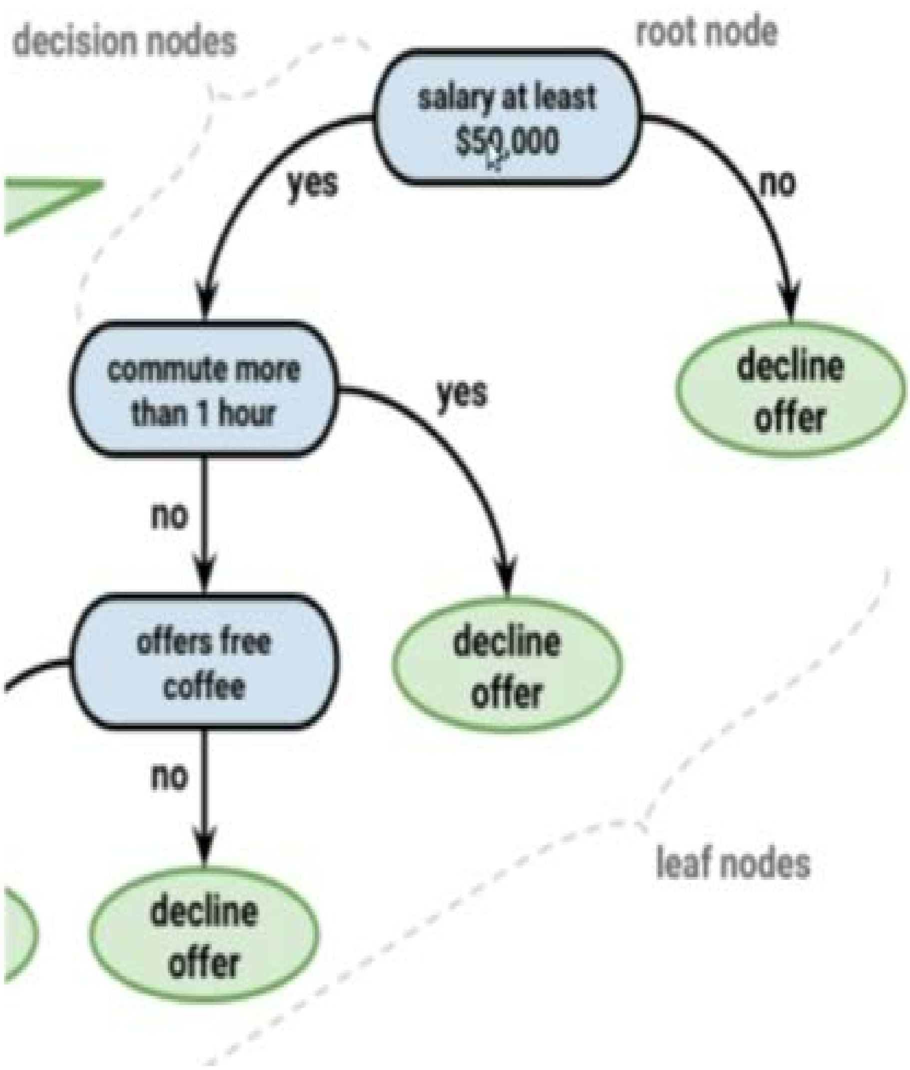


Decision Tree

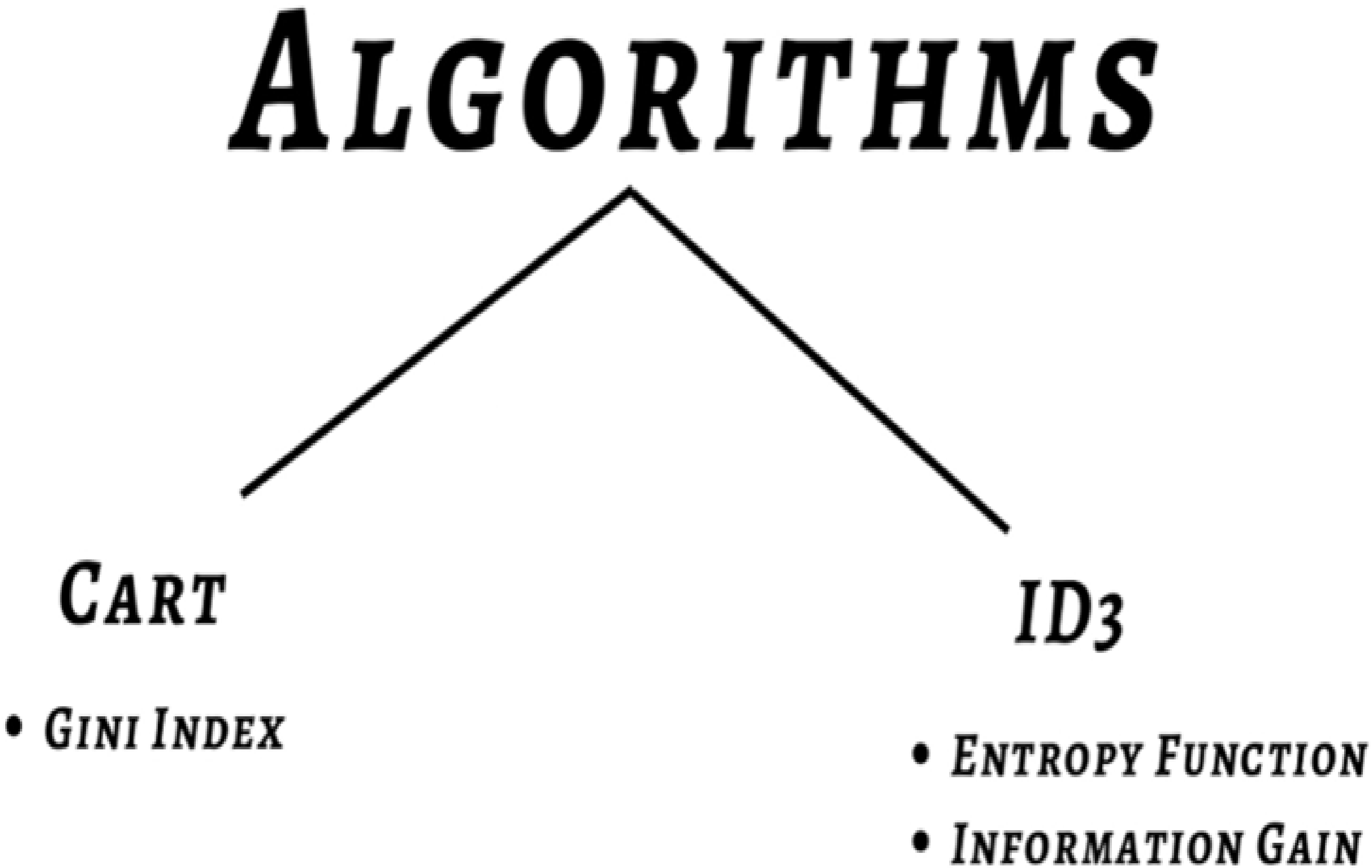


Decision Tree

A **Decision Tree** is a tree where *each node* represents a **Feature (Attribute)**, each link (branch) represents a **decision (rule)** and each leaf represents an **outcome**.



Decision Tree



Example: Decision Tree

S. No.	Outlook	Temperature	Humidity	Windy	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rainy	Mild	High	Weak	Yes
5	Rainy	Cool	Normal	Weak	Yes
6	Rainy	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rainy	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rainy	Mild	High	Strong	No

MAKE A DECISION TREE THAT PREDICTS WHETHER TENNIS WILL BE PLAYED ON THE DAY?

Example: Decision Tree

S. No.	Outlook	Temperature	Humidity	Windy	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rainy	Mild	High	Weak	Yes
5	Rainy	Cool	Normal	Weak	Yes
6	Rainy	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rainy	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rainy	Mild	High	Strong	No

Example: Decision Tree

Step 1: Create a root node

- How to choose the root node?

The attribute that best classifies the training data, use this attribute at the root of the tree.

- How to choose the best attribute?

So from here, *ID3 algorithm* begins

Example: Decision Tree

- Calculate **Entropy** (Amount of uncertainity in dataset):

$$Entropy = \frac{-p}{p+n}log_2(\frac{p}{p+n}) - \frac{n}{p+n}log_2(\frac{n}{p+n})$$

- Calculate **Average Information**:

$$I(Attribute) = \sum \frac{p_i + n_i}{p+n} Entropy(A)$$

- Calculate **Information Gain**: (Difference in Entropy before and after splitting dataset on attribute A)

$$Gain = Entropy(S) - I(Attribute)$$

Entropy: A measure of the disorder or randomness in a closed system

Example: Decision Tree

1. COMPUTE THE **ENTROPY** FOR DATA-SET **ENTROPY(S)**
2. FOR EVERY ATTRIBUTE/FEATURE:
    1. CALCULATE ENTROPY FOR ALL OTHER VALUES **ENTROPY(A)**
    2. TAKE **AVERAGE INFORMATION ENTROPY** FOR THE CURRENT ATTRIBUTE
    3. CALCULATE **GAIN** FOR THE CURRENT ATTRIBUTE
3. PICK THE **HIGHEST GAIN ATTRIBUTE**.
4. **REPEAT** UNTIL WE GET THE TREE WE DESIRED.

Example: Decision Tree

S. No.	Outlook	Temperature	Humidity	Windy	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rainy	Mild	High	Weak	Yes
5	Rainy	Cool	Normal	Weak	Yes
6	Rainy	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rainy	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rainy	Mild	High	Strong	No

P = 9

N = 5

Total = 14

Example: Decision Tree

- Calculate **Entropy(S):**

$$Entropy = \frac{-p}{p+n}log_2(\frac{p}{p+n}) - \frac{n}{p+n}log_2(\frac{n}{p+n})$$

$$Entropy(S) = \frac{-9}{9+5}log_2(\frac{9}{9+5}) - \frac{5}{9+5}log_2(\frac{5}{9+5})$$

$$Entropy(S) = \frac{-9}{14}log_2(\frac{9}{14}) - \frac{5}{14}log_2(\frac{5}{14}) \quad = 0.940$$

Try like this : log(9/14) / log(2)

Example: Decision Tree

- For each Attribute: (let say **Outlook**)
  - Calculate Entropy for each Values, i.e for 'Sunny', 'Rainy','Overcast'

Outlook	PlayTennis
Sunny	No
Sunny	No
Sunny	No
Sunny	Yes
Sunny	Yes

Outlook	PlayTennis
Rainy	Yes
Rainy	Yes
Rainy	No
Rainy	Yes
Rainy	No

Outlook	PlayTennis
Overcast	Yes
Overcast	Yes
Overcast	Yes
Overcast	Yes

Outlook	p	n	Entropy
Sunny	2	3	0.971
Rainy	3	2	0.971
Overcast	4	0	0

Example: Decision Tree

Calculate **Entropy(Outlook='Value'):**

$$Entropy = \frac{-p}{p+n}log_2(\frac{p}{p+n}) - \frac{n}{p+n}log_2(\frac{n}{p+n})$$

E (Outlook=sunny) = $-\frac{2}{5}\log\left(\frac{2}{5}\right)-\frac{3}{5}\log\left(\frac{3}{5}\right)=0.971$

E (Outlook=overcast) = $-1\log(1)-0\log(0)=0$

E (Outlook=rainy) = $-\frac{3}{5}\log\left(\frac{3}{5}\right)-\frac{2}{5}\log\left(\frac{2}{5}\right)=0.971$

Example: Decision Tree

- Calculate **Average Information Entropy**:

$$I(Outlook) = \frac{p_{sunny} + n_{sunny}}{p+n} Entropy(Outlook = Sunny) +$$

$$\frac{p_{rainy} + n_{rainy}}{p+n} Entropy(Outlook = Rainy) +$$

$$\frac{p_{Overcast} + n_{Overcast}}{p+n} Entropy(Outlook = Overcast)$$

$$I(Outlook) = \frac{3+2}{9+5} * 0.971 + \frac{2+3}{9+5} * 0.971 + \frac{4+0}{9+5} * 0 \quad = 0.693$$

Example: Decision Tree

- Calculate **Gain**: attribute is Outlook

$$Gain = Entropy(S) - I(Attribute)$$

$$Entropy(S) = 0.940$$

$$Gain(Outlook) = 0.940 - 0.693 = 0.247$$

Example: Decision Tree

- For each Attribute: (let say **Temperature**)
  - Calculate Entropy for each Temp, i.e for 'Hot', 'Mild' and 'Cool'

Temperature	PlayTennis
Hot	No
Hot	No
Hot	Yes
Hot	Yes

Temperature	PlayTennis
Mild	Yes
Mild	No
Mild	Yes
Mild	Yes
Mild	Yes
Mild	No

Temperature	PlayTennis
Cool	Yes
Cool	No
Cool	Yes
Cool	Yes

Temperature	p	n	Entropy
Hot	2	2	1
Mild	4	2	0.918
Cool	3	1	0.811

Example: Decision Tree

- Calculate **Average Information Entropy**:

$$I(Temperature) = \frac{p_{hot} + n_{hot}}{p+n} Entropy(Temperature = Hot) +$$

$$\frac{p_{mild} + n_{mild}}{p+n} Entropy(Temperature = Mild) +$$

$$\frac{p_{Cool} + n_{Cool}}{p+n} Entropy(Temperature = Cool)$$

$$I(Temperature) = \frac{2+2}{9+5} * 1 + \frac{4+2}{9+5} * 0.918 + \frac{3+1}{9+5} * 0.811 => 0.911$$

Example: Decision Tree

- Calculate **Gain**: attribute is Temperature

$$Gain = Entropy(S) - I(Attribute)$$

$$Entropy(S) = 0.940$$

$$Gain(Temperature) = 0.940 - 0.911 = 0.029$$

Example: Decision Tree

- For each Attribute: (let say **Humidity**)
  - Calculate Entropy for each Humidity, i.e for 'High', 'Normal'

Humidity	PlayTennis
Normal	Yes
Normal	No
Normal	Yes
Normal	Yes
Normal	Yes
Normal	Yes
Normal	Yes

Humidity	PlayTennis
High	No
High	No
High	Yes
High	Yes
High	No
High	Yes
High	No

Humidity	p	n	Entropy
High	3	4	0.985
Normal	6	1	0.591

Example: Decision Tree

- Calculate **Average Information Entropy**:

$$I(Humidity) = \frac{p_{High} + n_{High}}{p+n} Entropy(Humidity = High) +$$

$$\frac{p_{Normal} + n_{Normal}}{p+n} Entropy(Humidity = Normal)$$

$$I(Humidity) = \frac{3+4}{9+5} * 0.985 + \frac{6+1}{9+5} * 0.591 => 0.788$$

Example: Decision Tree

- Calculate **Gain**: attribute is Humidity

$$Gain = Entropy(S) - I(Attribute)$$

$$Entropy(S) = 0.940$$

$$Gain(Humidity) = 0.940 - 0.788 = 0.152$$

Example: Decision Tree

- For each Attribute: (let say **Windy**)
  - Calculate Entropy for each Windy, i.e for 'Strong' and 'Weak'

Windy	PlayTennis
Weak	No
Weak	Yes
Weak	Yes
Weak	Yes
Weak	No
Weak	Yes
Weak	Yes
Weak	Yes

Windy	PlayTennis
Strong	No
Strong	No
Strong	Yes
Strong	Yes
Strong	Yes
Strong	No

Windy	p	n	Entropy
Strong	3	3	1
Weak	6	2	0.811

Example: Decision Tree

- Calculate **Average Information Entropy**:

$$I(Windy) = \frac{p_{Strong} + n_{Strong}}{p+n} Entropy(Windy = Strong) +$$

$$\frac{p_{Weak} + n_{Weak}}{p+n} Entropy(Windy = Weak)$$

$$I(Windy) = \frac{3+3}{9+5} * 1 + \frac{6+2}{9+5} * 0.811 => 0.892$$

Example: Decision Tree

- Calculate **Gain**: attribute is Windy

$$Gain = Entropy(S) - I(Attribute)$$

$$Entropy(S) = 0.940$$

$$Gain(Windy) = 0.940 - 0.892 = 0.048$$

Example: Decision Tree

- Pick the highest gain attribute.

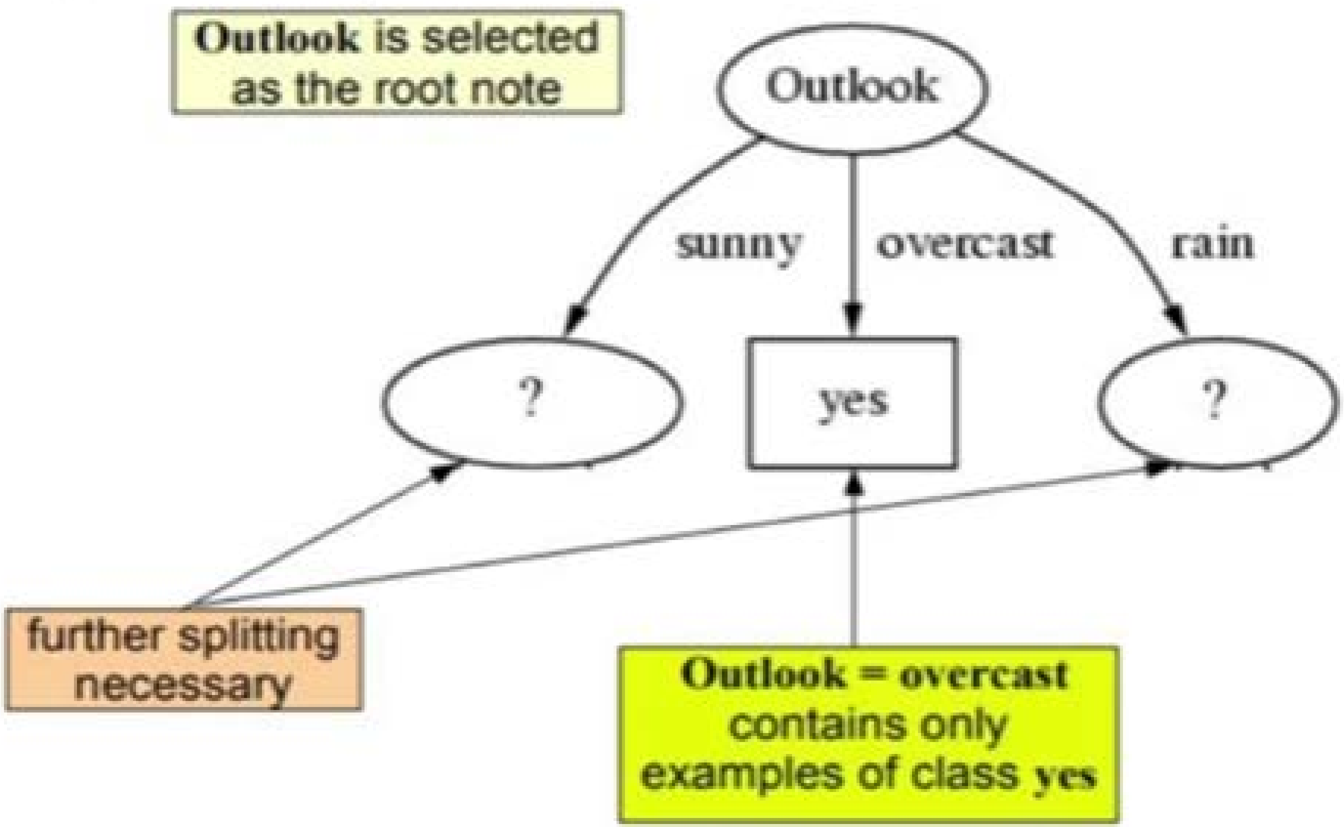
Attributes	Gain
Outlook	0.247
Temperature	0.029
Humidity	0.152
Windy	0.048

Root Node:

OUTLOOK

Example: Decision Tree

Outlook	Temperature	Humidity	Windy	PlayTennis
Overcast	Hot	High	Weak	Yes
Overcast	Cool	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes



Example: Decision Tree

- Repeat the same thing for sub-trees till we get the tree.

Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes

Outlook = "Sunny"

Outlook	Temperature	Humidity	Windy	PlayTennis
Rainy	Mild	High	Weak	Yes
Rainy	Cool	Normal	Weak	Yes
Rainy	Cool	Normal	Strong	No
Rainy	Mild	Normal	Weak	Yes
Rainy	Mild	High	Strong	No

Outlook = "Rainy"

Example: Decision Tree

Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes

P= 2 N= 3
Total= 5

- ENTROPY:

$$Entropy = \frac{-p}{p+n} log_2(\frac{p}{p+n}) - \frac{n}{p+n} log_2(\frac{n}{p+n})$$

$$Entropy(S_{sunny}) = \frac{-2}{2+3} log_2(\frac{2}{2+3}) - \frac{3}{2+3} log_2(\frac{3}{2+3})$$

=>0.971

Example: Decision Tree

- For each Attribute: (let say **Humidity**):
  - Calculate Entropy for each Humidity, i.e for 'High' and 'Normal'

Outlook	Humidity	PlayTennis
Sunny	High	No
Sunny	High	No
Sunny	High	No
Sunny	Normal	Yes
Sunny	Normal	Yes

Humidity	p	n	Entropy
high	0	3	0
normal	2	0	0

- Calculate **Average Information Entropy**: I(Humidity) = 0
- Calculate **Gain:** Gain = 0.971

Example: Decision Tree

- For each Attribute: (let say **Windy**):
  - Calculate Entropy for each Windy, i.e for 'Strong' and 'Weak'

Outlook	Windy	PlayTennis
Sunny	Strong	No
Sunny	Strong	Yes
Sunny	Weak	No
Sunny	Weak	No
Sunny	Weak	Yes

Windy	p	n	Entropy
Strong	1	1	1
Weak	1	2	0.918

- Calculate **Average Information Entropy**: I(Windy) = 0.951
- Calculate **Gain:** Gain = 0.020

Example: Decision Tree

- For each Attribute: (let say **Temperature**):
  - Calculate Entropy for each Windy, i.e for 'Cool', 'Hot' and 'Mild'

Outlook	Temperature	PlayTennis
Sunny	Cool	Yes
Sunny	Hot	No
Sunny	Hot	No
Sunny	Mild	No
Sunny	Mild	Yes

Temperature	p	n	Entropy
Cool	1	0	0
Hot	0	2	0
Mild	1	1	1

- Calculate **Average Information Entropy**: I(Temp) = 0.4

- Calculate **Gain:** Gain = 0.571

Example: Decision Tree

- Pick the highest gain attribute.

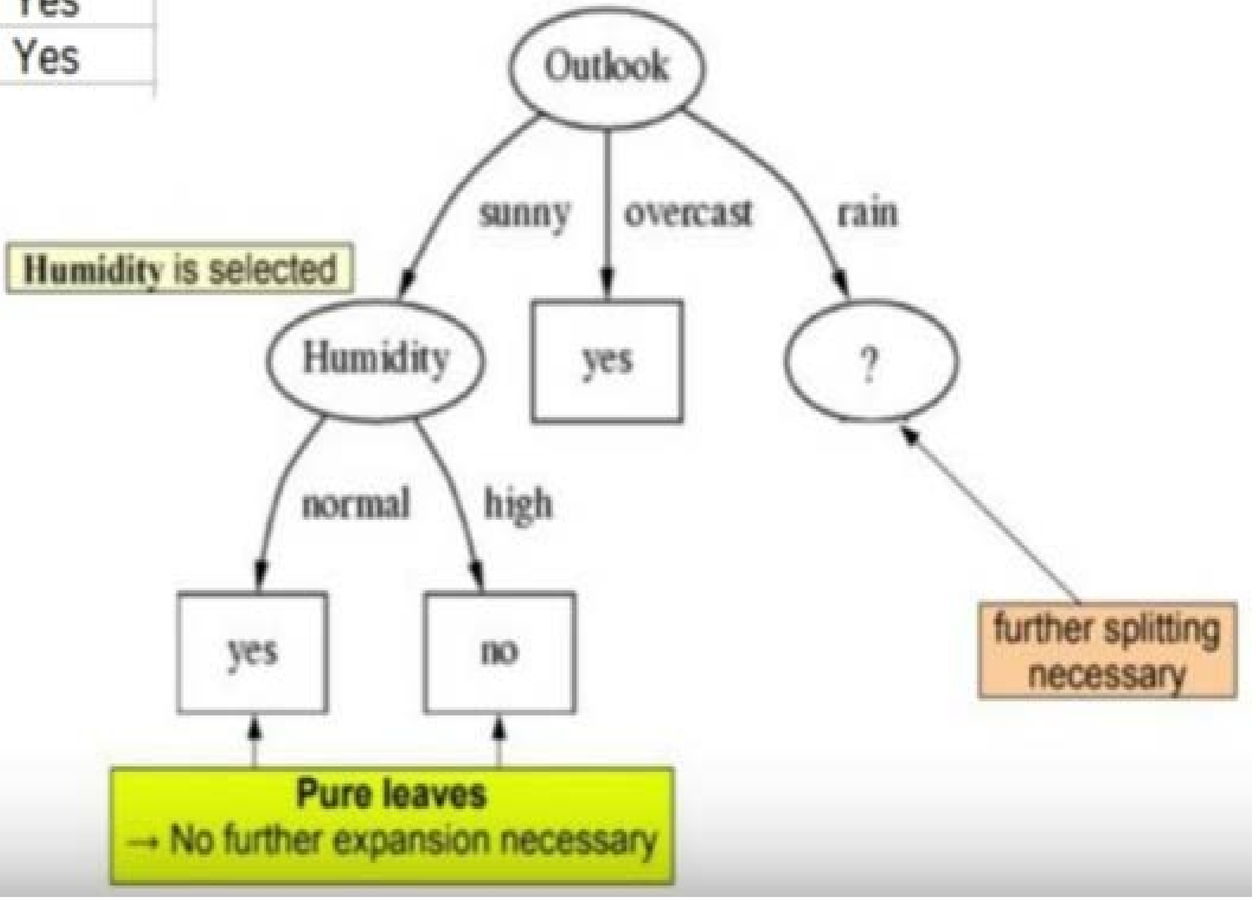
Attributes	Gain
Temperature	0.571
Humidity	0.971
Windy	0.02

Next Node in sunny:

Humidity

Example: Decision Tree

Outlook	Humidity	PlayTennis
Sunny	High	No
Sunny	High	No
Sunny	High	No
Sunny	Normal	Yes
Sunny	Normal	Yes



Example: Decision Tree

Outlook	Temperature	Humidity	Windy	PlayTennis
Rainy	Mild	High	Weak	Yes
Rainy	Cool	Normal	Weak	Yes
Rainy	Cool	Normal	Strong	No
Rainy	Mild	Normal	Weak	Yes
Rainy	Mild	High	Strong	No

P= N=

3 2

Total = 5

- ENTROPY:

$$Entropy = \frac{-p}{p+n}log_2(\frac{p}{p+n}) - \frac{n}{p+n}log_2(\frac{n}{p+n})$$

$$Entropy(S_{Rainy}) = \frac{-3}{3+2}log_2(\frac{3}{3+2}) - \frac{2}{3+2}log_2(\frac{2}{2+3})$$

=>0.971

Example: Decision Tree

- For each Attribute: (let say **Humidity**):
  - Calculate Entropy for each Humidity, i.e for 'High' and 'Normal'

Outlook	Humidity	PlayTennis
Rainy	High	Yes
Rainy	High	No
Rainy	Normal	Yes
Rainy	Normal	No
Rainy	Normal	Yes

Attribute	p	n	Entropy
High	1	1	1
Normal	2	1	0.918

  - Calculate **Average Information Entropy**: I(Humidity) = 0.951
  - Calculate **Gain:** Gain = 0.020

Example: Decision Tree

- For each Attribute: (let say **Windy**):
  - Calculate Entropy for each Windy, i.e for 'Strong' and 'Weak'

Outlook	Windy	PlayTennis
Rainy	Strong	No
Rainy	Strong	No
Rainy	Weak	Yes
Rainy	Weak	Yes
Rainy	Weak	Yes

Attribute	p	n	Entropy
Strong	0	2	0
Weak	3	0	0

  - Calculate **Average Information Entropy**: I(Windy) = 0
  - Calculate **Gain:** Gain = 0.971

Example: Decision Tree

- For each Attribute: (let say **Temperature**):
  - Calculate Entropy for each Windy, i.e for 'Cool', 'Hot' and 'Mild'

Outlook	Temperature	PlayTennis
Rainy	Mild	Yes
Rainy	Cool	Yes
Rainy	Cool	No
Rainy	Mild	Yes
Rainy	Mild	No

Attribute	p	n	Entropy
Cool	1	1	1
Mild	2	1	0.918

  - Calculate **Average Information Entropy**: I(Temp) = 0.951
  - Calculate **Gain:** Gain = 0.020

Example: Decision Tree

- Pick the highest gain attribute.

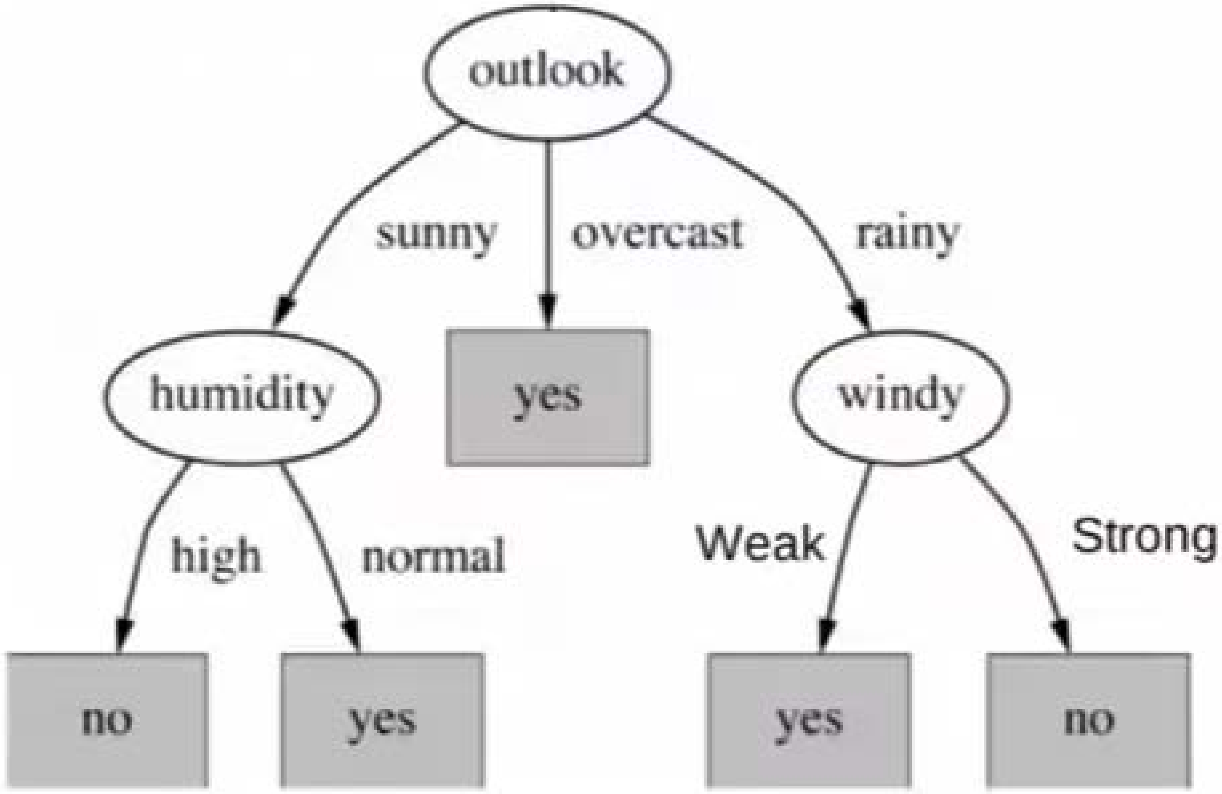
Attributes	Gain
Humidity	0.02
Windy	0.971
Temperature	0.02

Next Node in

Rainy: **Windy**

Example: Decision Tree

Final decision tree



Decision Tree pros and cons

Advantages:

- Easy to understand and interpret, perfect for visual representation.
- Requires little data preprocessing.
- Non-parametric model: no assumptions about the shape of data.
- Feature selection happens automatically: unimportant features will not influence the result. The presence of features that depend on each other also doesn't affect the quality.

Disadvantages:

- They are unstable, meaning that a small change in the data can lead to a large change in the structure of the optimal decision tree.
- They are often relatively inaccurate. Many other predictors perform better with similar data. This can be remedied by replacing a single decision tree with a random forest of decision trees, but a random forest is not as easy to interpret as a single decision tree.
- Calculations can get very complex, particularly if many values are uncertain and/or if many outcomes are linked.

Decision Tree: Problem and Solution

Predictors				Target
Outlook	Temp.	Humidity	Windy	Hours Played
Rainy	Hot	High	False	26
Rainy	Hot	High	True	30
Overcast	Hot	High	False	48
Sunny	Mild	High	False	46
Sunny	Cool	Normal	False	62
Sunny	Cool	Normal	True	23
Overcast	Cool	Normal	True	43
Rainy	Mild	High	False	36
Rainy	Cool	Normal	False	38
Sunny	Mild	Normal	False	48
Rainy	Mild	Normal	True	48
Overcast	Mild	High	True	62
Overcast	Hot	Normal	False	44
Sunny	Mild	High	True	30

Decision Tree: Problem and Solution

a) Standard deviation for **one** attribute:

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

$$Count = n = 14$$

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

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

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

- Standard Deviation (**S**) is for tree building (branching).
- Coefficient of Deviation (**CV**) is used to decide when to stop branching. We can use Count (**n**) as well.
- Average (**Avg**) is the value in the leaf nodes.

Decision Tree: Problem and Solution

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

Total count = 14

Average = (25 + 30 + 46 + 45 + 52 + 23 + 43 + 35 + 38 + 46 + 48 + 52 + 44 + 30)/14 = 39.7857 = 39.79

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

Let $A = \sum(x - \bar{x})^2 = (25 - 39.8)^2 + (30 - 39.8)^2 + (46 - 39.8)^2 + (45 - 39.8)^2$

$+ (52 - 39.8)^2 + (23 - 39.8)^2 + (43 - 39.8)^2 + (35 - 39.8)^2 + (38 - 39.8)^2$

$+ (46 - 39.8)^2 + (48 - 39.8)^2 + (52 - 39.8)^2 + (44 - 39.8)^2 + (30 - 39.8)^2$

$= 219.04 + 96.04 + 38.44 + 27.04 + 148.84 + 282.24 + 10.24 + 23.04$

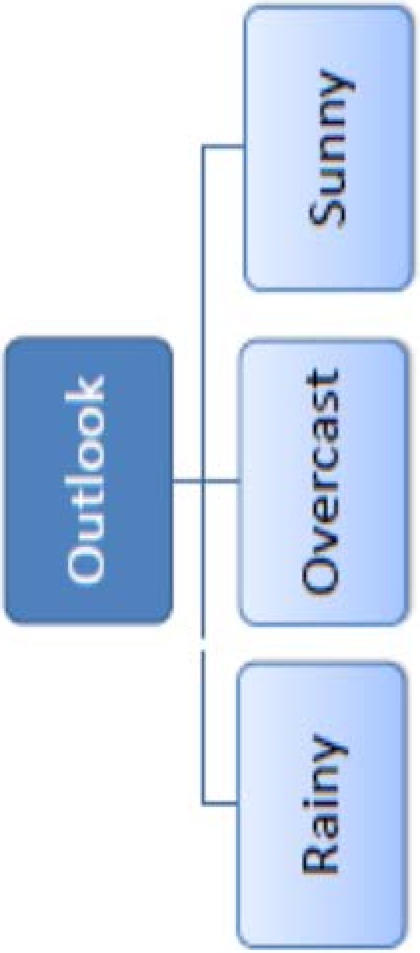
$+ 3.24 + 38.44 + 67.24 + 148.84 + 17.64 + 96.04$

$= 1216.36$

$$S = \sqrt{\frac{A}{n}} = \sqrt{\frac{1216.36}{14}} = \sqrt{86.88} = 9.32$$

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

Decision Tree: Problem and Solution



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

Decision Tree: Problem and Solution

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

Consider Sunny:

Total count = 5

Average = (45 + 52 + 23 + 46 + 30) / 5 = 39.2

Let $A = \sum(x - \bar{x})^2 = (45 - 39.2)^2 + (52 - 39.2)^2 + (23 - 39.2)^2$

$+ (46 - 39.2)^2 + (30 - 39.2)^2 = 590.8$

$$S = \sqrt{\frac{A}{n}} = \sqrt{\frac{590.8}{5}} = \sqrt{118.16} = 10.87$$

Consider Overcast:

Total count = 4

Average = (46 + 43 + 52 + 44) / 4 = 46.25

Let $A = \sum(x - \bar{x})^2 = (46 - 46.25)^2 + (43 - 46.25)^2 + (52 - 46.25)^2$

$+ (44 - 46.25)^2 = 48.75$

$$S = \sqrt{\frac{A}{n}} = \sqrt{\frac{48.75}{4}} = \sqrt{12.1875} = 3.49$$

Consider Rainy:

Total count = 5

Average = (25 + 30 + 35 + 38 + 48) / 5 = 35.2

Let $A = \sum(x - \bar{x})^2 = (25 - 35.2)^2 + (30 - 35.2)^2 + (35 - 35.2)^2$

$+ (38 - 35.2)^2 + (48 - 35.2)^2 = 302.8$

$$S = \sqrt{\frac{A}{n}} = \sqrt{\frac{302.8}{5}} = \sqrt{60.56} = 7.78$$

Decision Tree: Problem and Solution

b) Standard deviation for **two** attributes (target and predictor):

$$S(T,X)=\sum_{c\in X}P(c)S(c)$$

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

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

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

= 7.66

Decision Tree: Problem and Solution

Standard Deviation Reduction

The standard deviation reduction is based on the decrease in standard deviation after a dataset is split on an attribute. Constructing a decision tree is all about finding attribute that returns the highest standard deviation reduction (i.e., the most homogeneous branches).

Step 1: The standard deviation of the target is calculated.

Standard deviation (Hours Played) = 9.32

Decision Tree: Problem and Solution

Step 2: The dataset is then split on the different attributes. The standard deviation for each branch is calculated. The resulting standard deviation is subtracted from the standard deviation before the split. The result is the standard deviation reduction.

		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=0.17		

		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

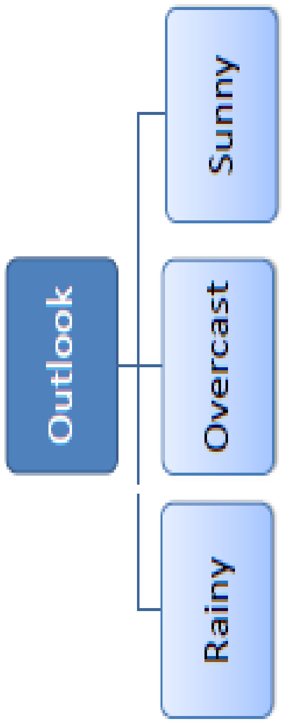
Decision Tree: Problem and Solution

Step 3: The attribute with the largest standard deviation reduction is chosen for the decision node.

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

Decision Tree: Problem and Solution

Step 4a: The dataset is divided based on the values of the selected attribute. This process is run recursively on the non-leaf branches, until all data is processed.



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

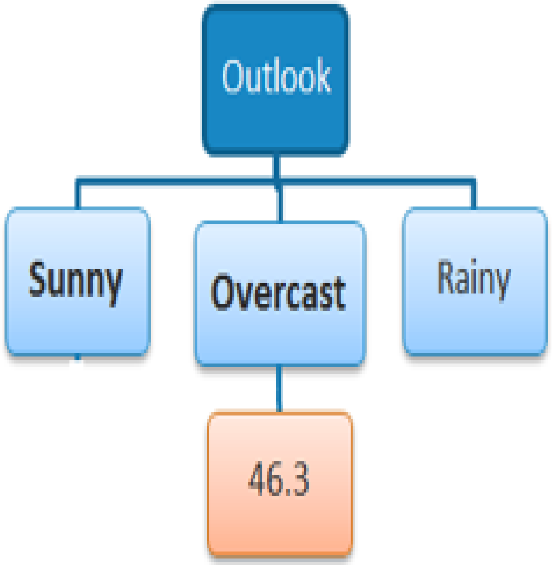
In practice, we need some termination criteria. For example, when coefficient of deviation (**CV**) for a branch becomes smaller than a certain threshold (e.g., 10%) and/or when too few instances (**n**) remain in the branch (e.g., 3).

Decision Tree: Problem and Solution

Step 4b: "Overcast" subset does not need any further splitting because its CV (8%) is less than the threshold (10%). 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



Decision Tree: Problem and Solution

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 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

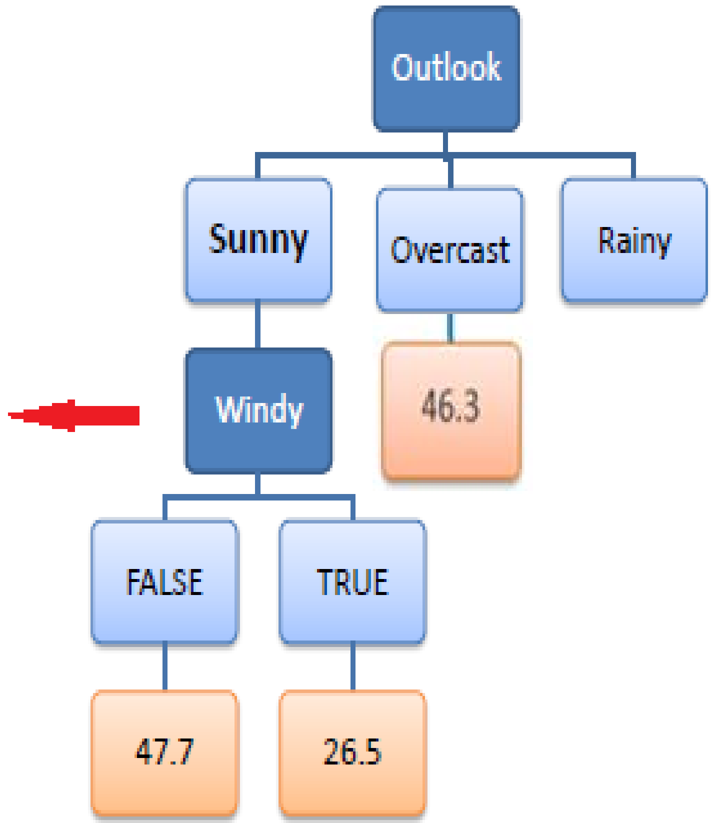
		Hours Played (StDev)	Count
Windy	False	3.09	3
	True	3.50	2

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

Decision Tree: Problem and Solution

Because the number of data points for both branches (FALSE and TRUE) is equal or 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



Decision Tree: Problem and Solution

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

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

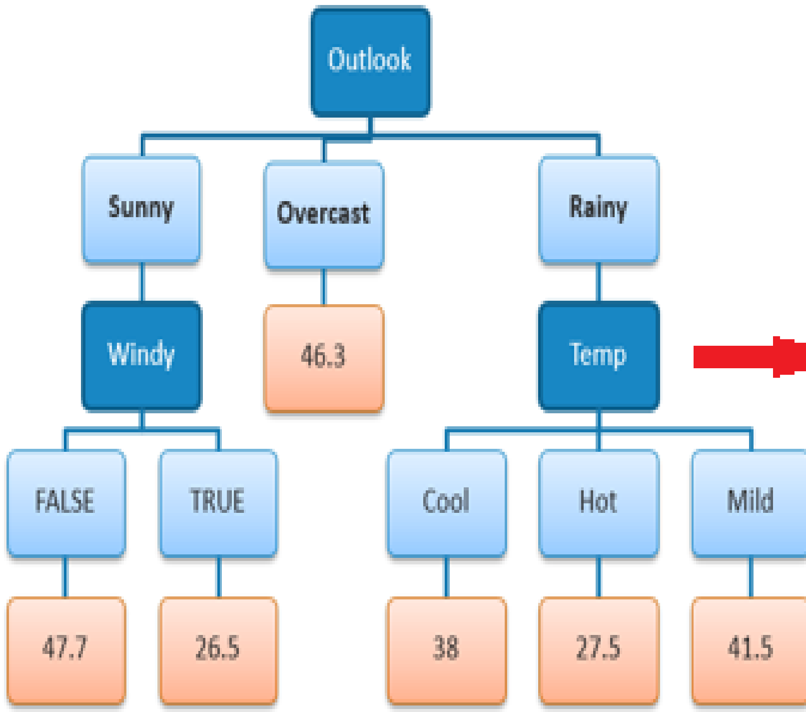
SDR = 7.78 - ((3/5)*4.1 + (2/5)*5.0) = **3.32**

		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**

Decision Tree: Problem and Solution

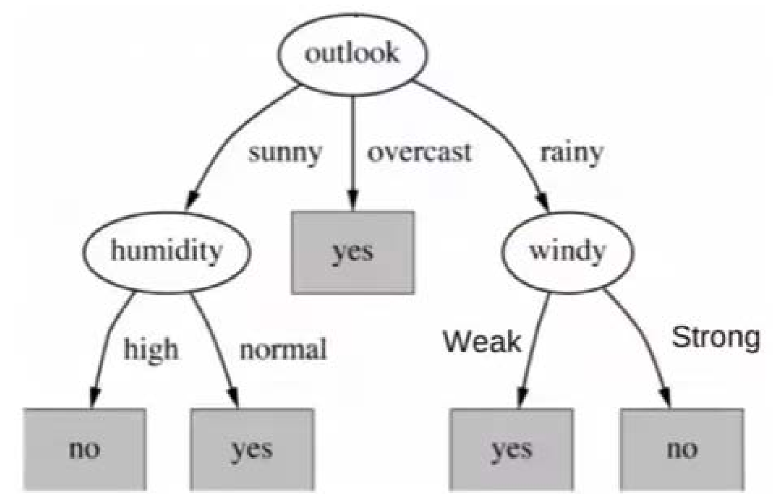
Because the number of data points for all three branches (Cool, Hot and Mild) is equal or less than 3 we stop further branching and assign the average of each branch to the related leaf node.

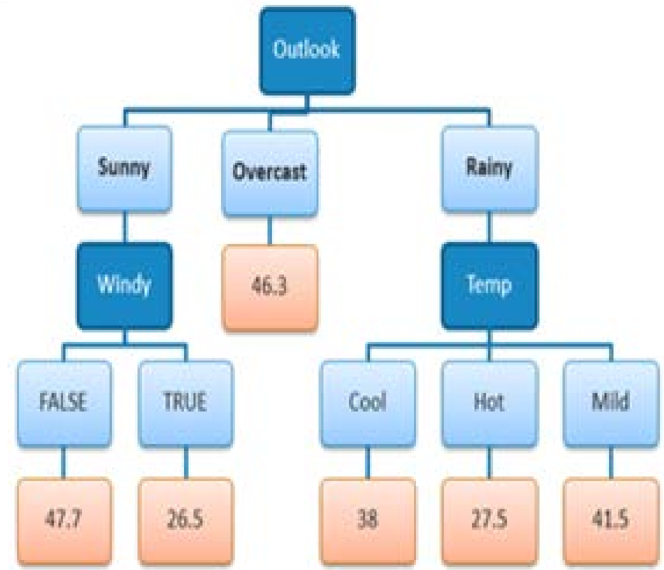


Temp	Hours Played
Cool	38
Hot	25
Hot	30
Mild	35
Mild	48

When the number of instances is more than one at a *leaf node* we calculate the *average* as the final value for the target.

Final decision tree





Thanks !!!