SE488 Ai Driven Software Development

Learning Algorithms

Outline

- Introduction
- Types of AI
- Algorithms in AI
- Learning Algorithms :
  - Decision Tree

Types of AI

- Narrow AI
- General AI – AGI (Artificial General Intelligence)
- Superintelligent AI

Type of AI :Narrow AI

- Narrow AI, also known as Weak AI, represents AI systems that are designed to perform a specific task, and their functionality doesn't extend beyond that specific task

- Narrow AI focuses on a single task and is constrained by constraints to not go beyond that, leaving it unable to solve unfamiliar problems.

- Weak refers to specialization not to capability

Type of AI :Narrow AI (examples)

- Personal Assistants :They use natural language processing and voice recognition technologies to understand and respond to user commands.

- Recommendation systems : are another prevalent example of Narrow AI. These systems employ machine learning algorithms to analyze user behavior and preferences, predict potential interests, and recommend relevant items or content.

- Autonomous Vehicles : represent a complex and advanced application of Narrow AI. Companies like Tesla have developed self-driving cars that use a combination of computer vision, sensor technology, and machine learning to navigate roads, interpret traffic signs, and avoid obstacles.

Type of AI :General AI – AGI (Artificial General Intelligence)

- General AI can manage to perform a broad range of tasks by using human-like cognitive capabilities
- Strong AI, allows machines to apply knowledge and skills in different contexts.
- The objective of AGI is to create machines that can reason and think just like a human is capable of doing.

Narrow AI Vs General AI

Narrow AI	General AI
○ Application specific/ task limited	○ Perform general (human) intelligent action
○ Fixed domain models provided by programmers	○ Self-learns and reasons with its operating environment
○ Learns from thousands of labeled examples	○ Learns from few examples and/or from unstructured data
○ Reflexive tasks with no understanding	○ Full range of human cognitive abilities
○ Knowledge does not transfer to other domains or tasks	○ Leverages knowledge transfer to new domains and tasks

Type of AI :Superintelligent AI

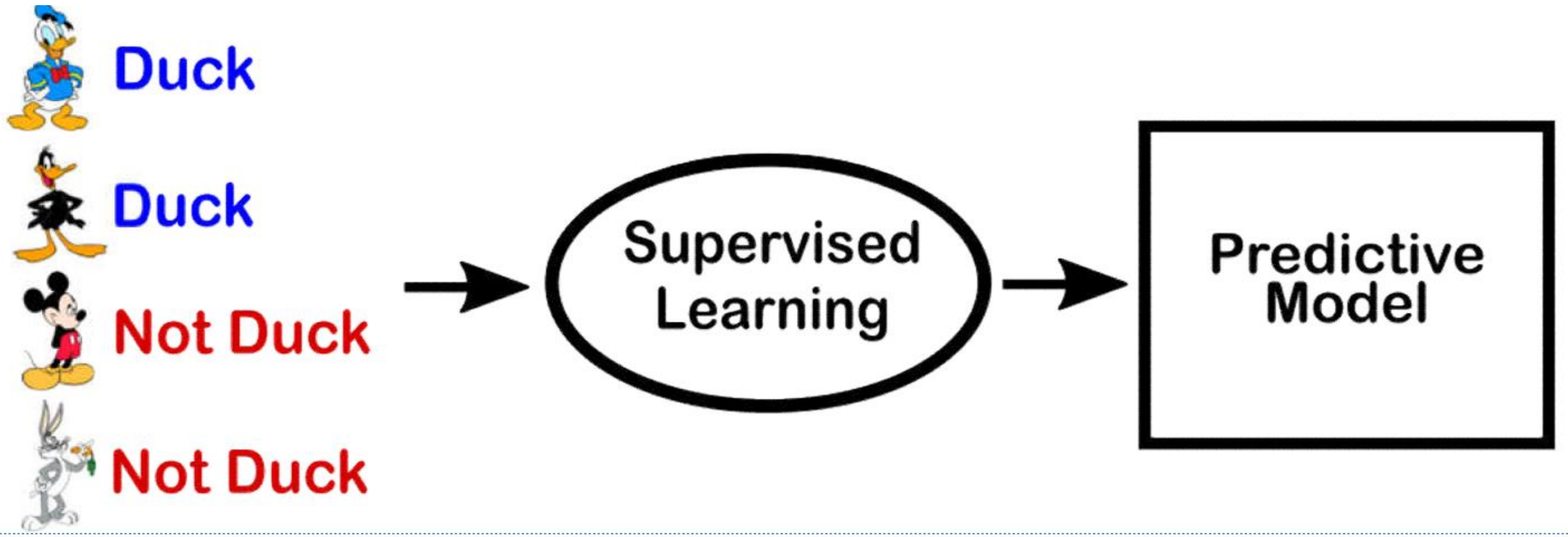
- Super AI can be able to outperform human intelligence.
- Solve problems like : climate change, disease, poverty etc.
- Extinction of the human race ?

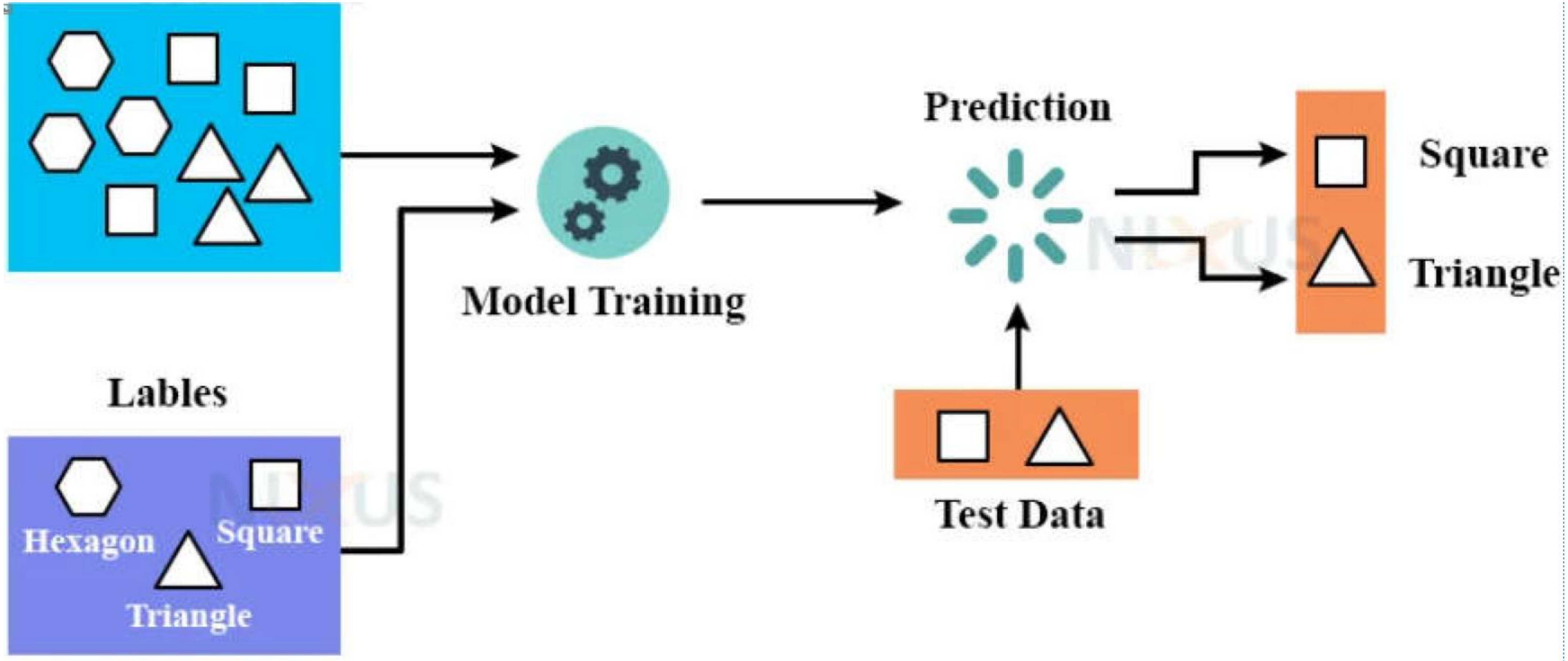
Algorithms in AI

- Deterministic, Terminating, Feasible
- Optimization, Automation, Innovation
- Types :
  - Supervised Learning Algorithms
  - Unsupervised Learning Algorithms
  - Reinforcement Learning Algorithms

Supervised Learning Algorithm

- Supervised learning is a crucial part of machine learning and it's all about making educated predictions based on data that has already been labelled manually.





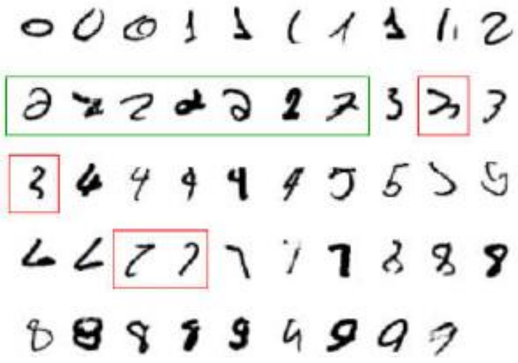
Source : https://nixustechnologies.com/

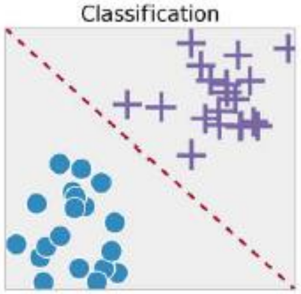
Examples Supervised Learning Algorithm

- *Linear Regression:* Think of this as an algorithm that helps us guess future values based on past trends. For instance, if we know the size, location, and age of several houses along with their prices, linear regression helps us predict how much a new house might cost based on these features. It does this by finding the best-fitting line, known as the regression line
- *Logistic Regression:* This is actually about classifying things into categories. For instance, it could be used to decide whether an email is spam (labeled as 1) or not spam (labeled as 0) based on the email's content, sender, and so on.
- *Support Vector Machines*: SVM used for categorizing things, even when the categories aren't easily separated. This could be classifying emails as 'business' or 'personal', even when there's a lot of overlap between the two. It works by finding a boundary (the hyperplane) that best divides the categories based on the data.

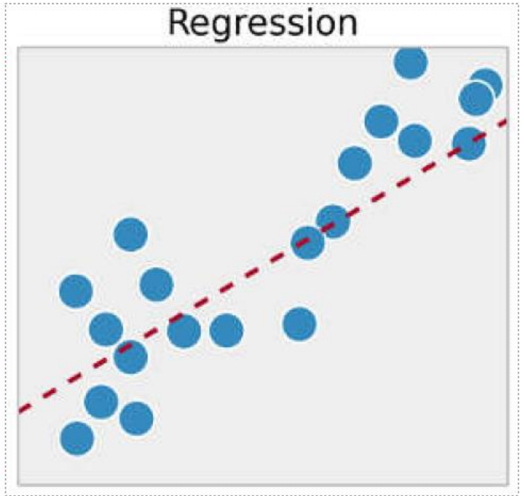
Outlook	Temperature	Humidity	Wind	PlayTennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes

Source : Machine Learning, Chapter 3, Tom M. Mitchell, McGraw-Hill, 1997



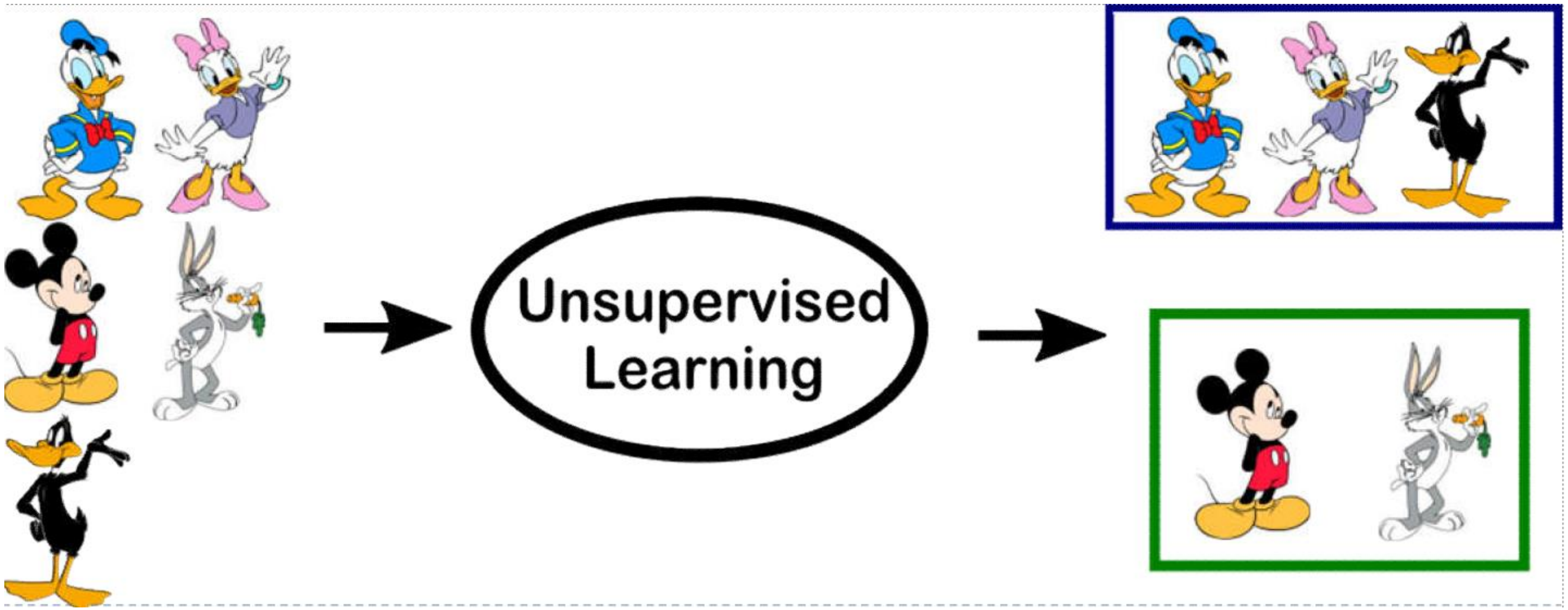


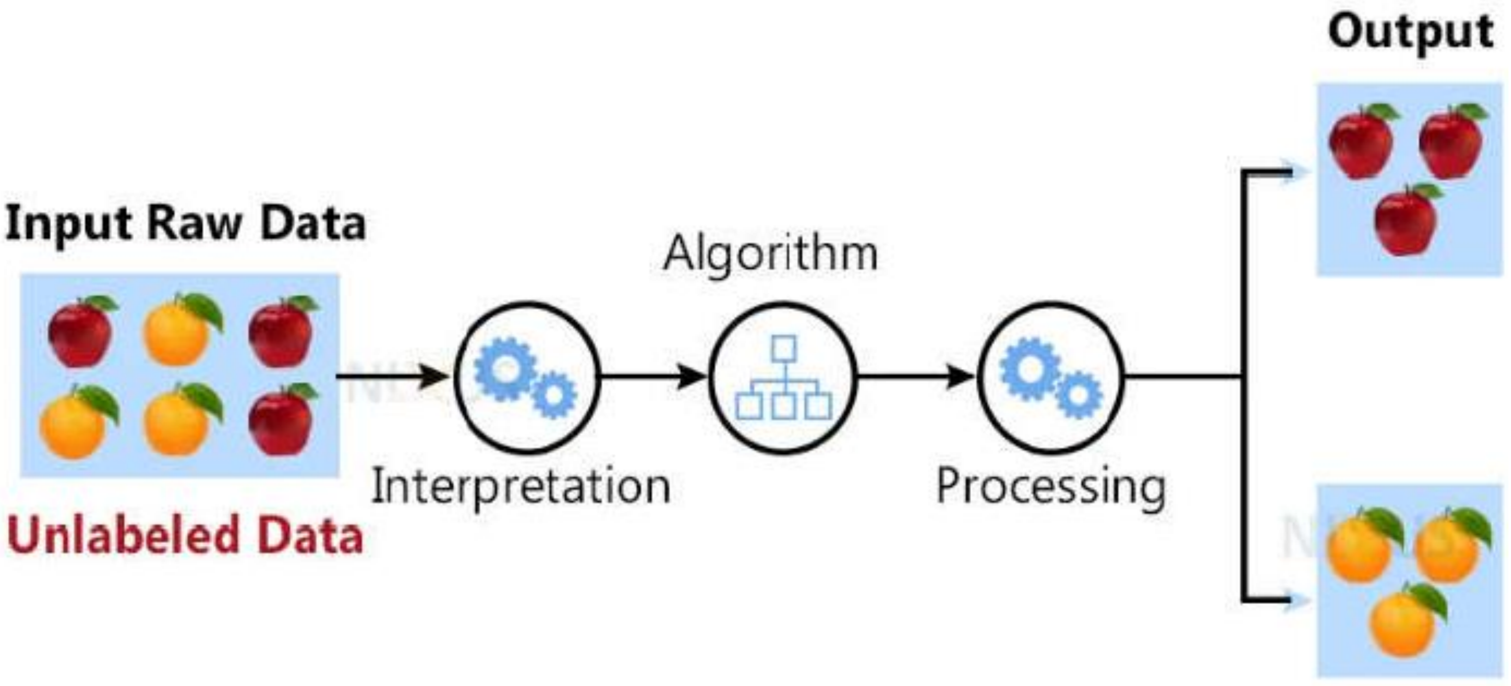
R&D Spend	Administration	Marketing Spend	State	Profit
165349.2	136897.8	471784.1	New York	192261.83
162597.7	151377.59	443898.53	California	191792.06
153441.51	101145.55	407934.54	Florida	191050.39
144372.41	118671.85	383199.62	New York	182901.99
142107.34	91391.77	366168.42	Florida	166187.94
131876.9	99814.71	362861.36	New York	156991.12



Unsupervised Learning Algorithms

- Unsupervised learning is about understanding, structuring, and extracting meaningful information from unlabeled data.



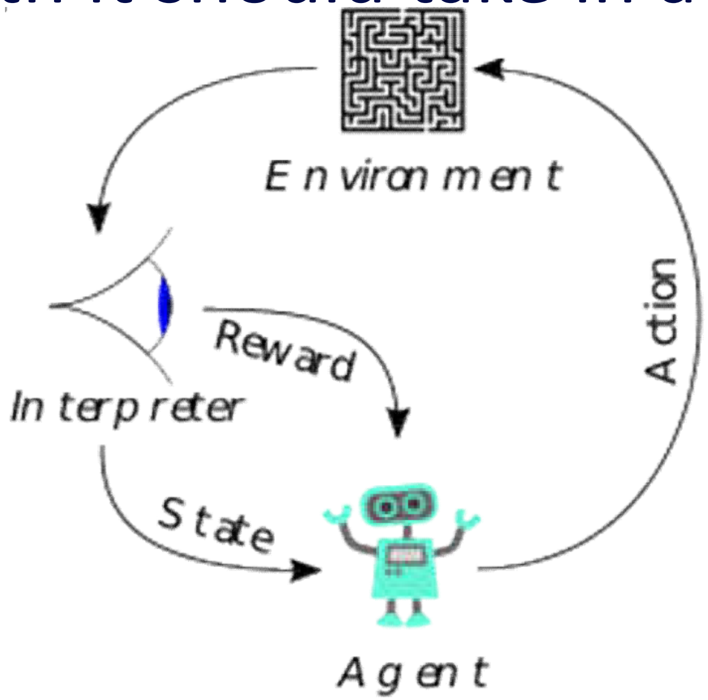


Source : https://nixustechnologies.com/

- *K-means clustering:* Imagine you're a librarian with a pile of books that haven't been sorted by genre. K-means clustering is an algorithm that helps you sort these books into distinct groups based on common attributes like book title, author, or even the first sentence. Similarly, this algorithm can help businesses understand their customers better by grouping them based on their purchasing behaviors.
- *Principal Component Analysis (PCA):* Sometimes, we have more data than we know what to do with. PCA is a technique that reduces this data to a manageable size, while keeping its important structure and relationships intact. It's like summarizing a lengthy book into a few key points that still capture the essence of the story.

Reinforcement Learning Algorithms

- Reinforcement learning is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behavior or path it should take in a specific context.



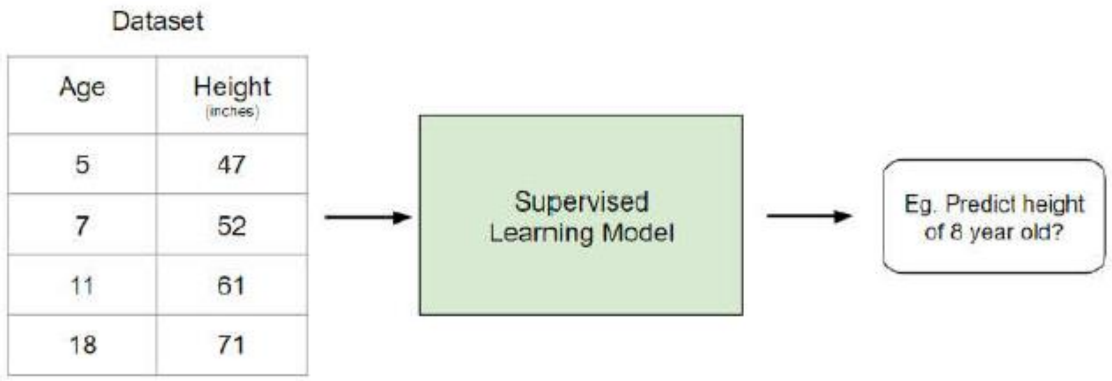
Source : https://techvidvan.com/

- *Q-learning*: It's a strategy-free reinforcement learning algorithm. Imagine teaching a robot to navigate a maze. The robot tries different paths (actions), and each time it hits a dead-end or finds the exit, it learns more about the maze (current state). Over time, the robot learns the most efficient way to navigate the maze.
- Deep Q Network (DQN): This is an advanced form of Q-learning that works well with more complex problems. It combines Q-learning with deep neural networks to handle large and complicated situations, like playing video games at superhuman levels.

Outline

- Supervised Learning Algorithm
  - Decision Tree

- Supervised learning, also known as supervised machine learning, is a subcategory of machine learning and artificial intelligence. It is defined by its use of labeled datasets to train algorithms that to classify data or predict outcomes accurately. Citation :IBM

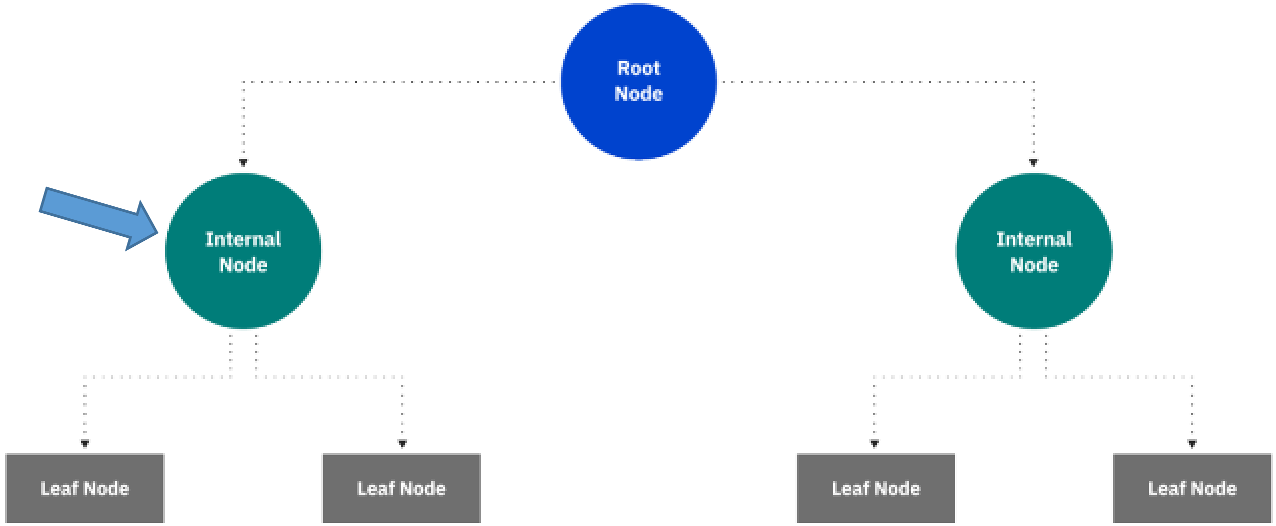


Decision Tree

- A decision tree is a tree-like structure that is used as a model for classifying data.
- A decision tree is made up of three types of nodes
  - *Decision Nodes*: These type of node have two or more branches
  - *Leaf Nodes*: The lowest nodes which represents decision
  - *Root Node*: This is also a decision node but at the topmost level

Decision Tree

Or Branches

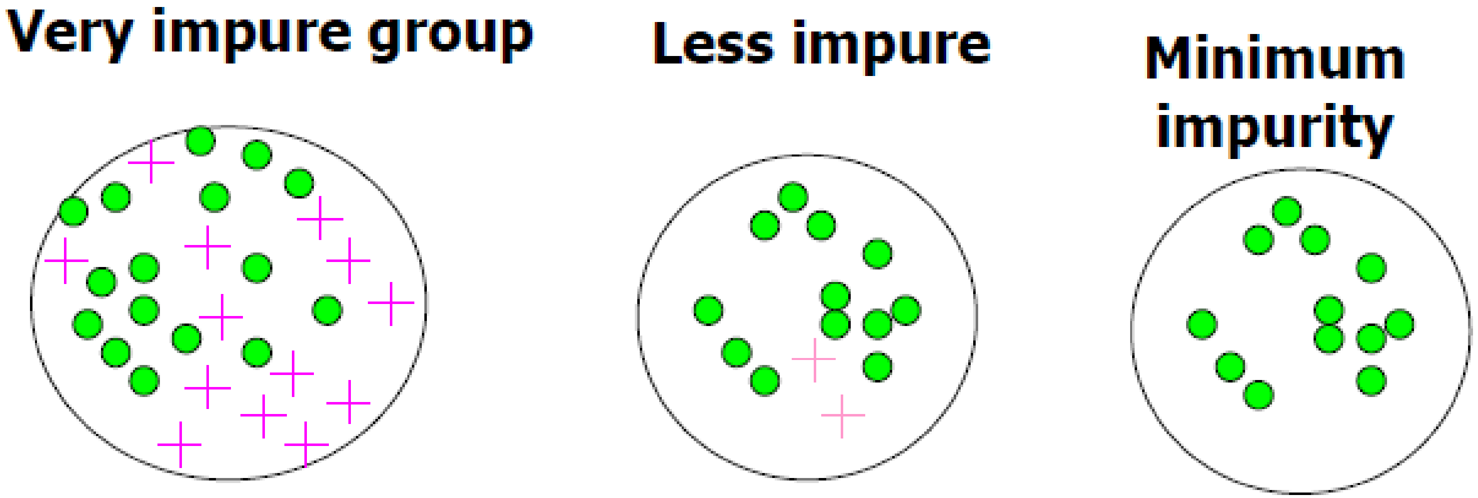


Steps to Build a Decision Tree

- Determine the Decision Column (Target Variable) and Identify the Features (Input Variables)
- Determine the Best Split Criteria
  - Calculate the Entropy for the classes
  - Calculate the Entropy for Other Attributes (After potential Split)
  - Calculate the Information Gain (for Each potential Split)
- Perform the First Split
- Perform Further Splits (if Any)

Terminology

- **Entropy**
  - Entropy is a measure of uncertainty or randomness in the data. In the context of decision trees, it quantifies how pure or impure a set of data is in terms of the class labels.
  - When all instances in a set belong to one class, entropy is 0 (pure).



- For a dataset that has C classes and the probability of choosing data from class, i is Pi. Then entropy E(S) can be mathematically represented as :

$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

•*S* is the set of instances.
•*pi* is the proportion of instances belonging to class i in the set S.
•c is the number of classes

Example

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

E = ?

E = ?

Entropy Calculation Examples

- If all the 10 observations belong to 1 class
- If both classes YES and NO have an equal number of observations
- If we have a dataset of 10 observations belonging to two classes YES and NO. If 6 observations belong to the class, YES, and 4 observations belong to class NO, then

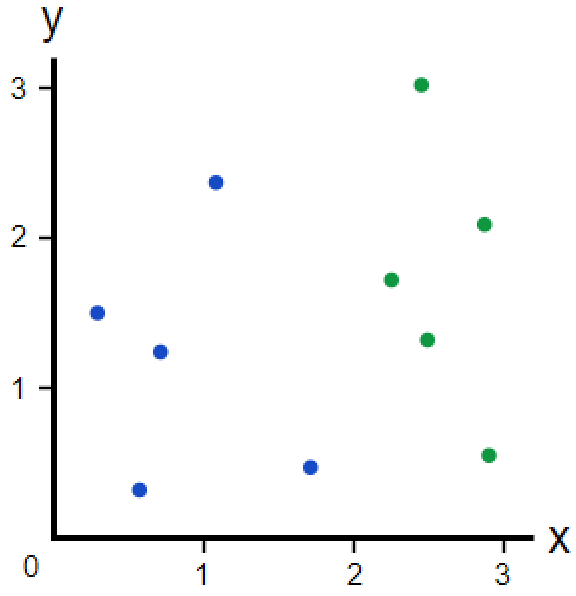
Information Gain

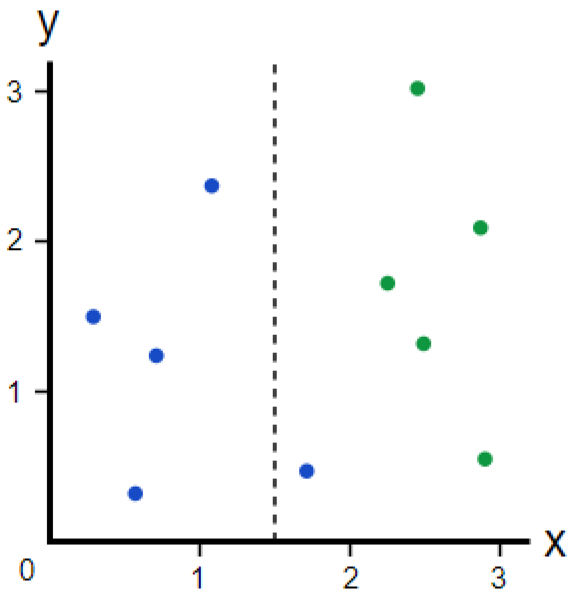
- How can we quantify the quality of a split ?
  - Information gain is a metric used to choose the best feature to split the data.
  - Information gain : subtract the weighted entropies of each branch from the original entropy. the best split is chosen by maximizing Information Gain.

$$IG(Y,X) = E(Y) - E(Y|X)$$

$$\sum P(c)E(c)$$

Example 1

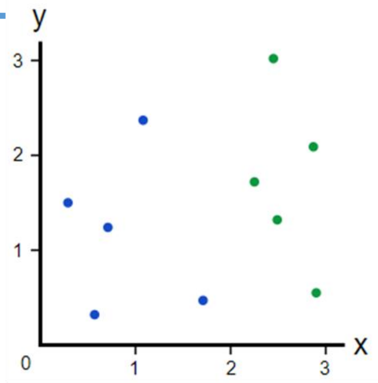


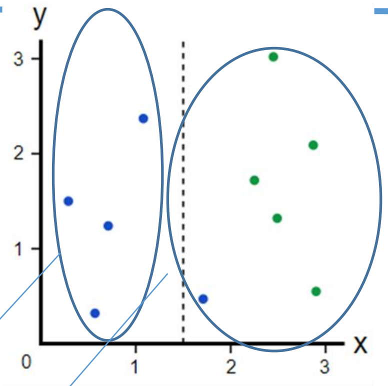


$$E_{before} = -(0.5\log_2 0.5 + 0.5\log_2 0.5) \\ = \boxed{1}$$

$$E_{left} = \boxed{0}$$

$$E_{right} = -(\frac{1}{6}\log_2(\frac{1}{6}) + \frac{5}{6}\log_2(\frac{5}{6})) \\ = \boxed{0.65}$$





$$E_{before} = -(0.5 \log_2 0.5 + 0.5 \log_2 0.5) \\ = \boxed{1}$$

$$E_{left} = \boxed{0}$$

$$E_{right} = -(\frac{1}{6}\log_2(\frac{1}{6}) + \frac{5}{6}\log_2(\frac{5}{6})) \\ = \boxed{0.65}$$

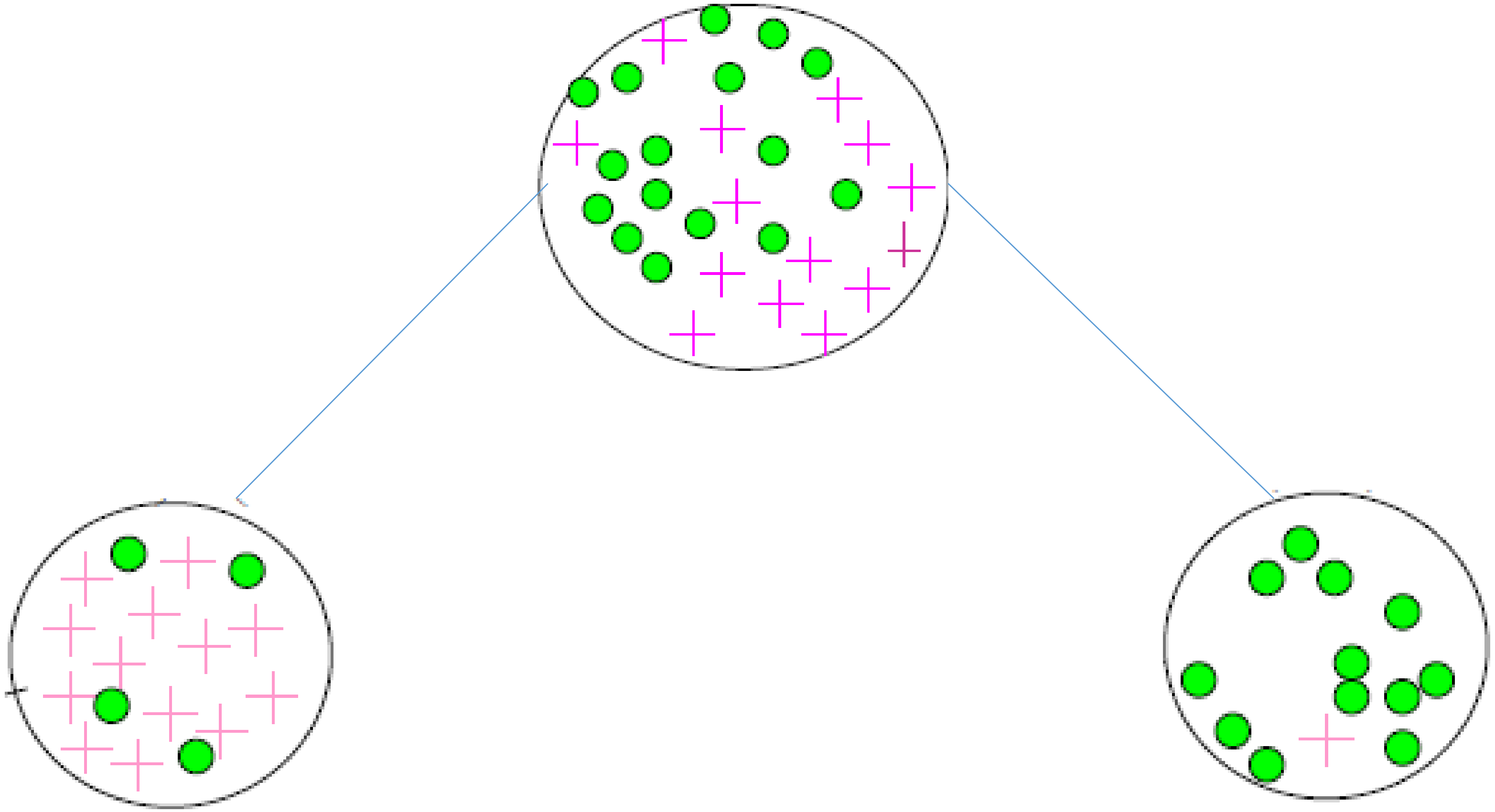
- we have the entropies for both branches.
- we can determine the quality of the split by **weighting the entropy of each branch by how many elements it has**.

$$E_{split} = 0.4 * 0 + 0.6 * 0.65 \\ = \boxed{0.39}$$

- Information gain :

$$\text{Gain} = 1 - 0.39 = \boxed{0.61}$$

Example 2



Build the Tree

- The decision tree is built by recursively selecting the feature with the highest information gain (or, equivalently, the greatest reduction in entropy) at each node to split the data. This process continues until the data is fully classified, or another stopping criterion (like tree depth or minimum node size) is met.

DataSet

Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No

Step by Step Procedure for Building a Decision Tree

- **Step 1**: Determine the Decision Column and features (attributes)
- Since decision trees are used for clasification, you need to determine the classes which are the basis for the decision.

- To determine the rootNode we need to compute the entropy.

Play Golf(14)	
Yes	No
9	5

- Step 2: Calculating Entropy for the classes (Play Golf)
- In this step, you need to calculate the entropy for the Play Golf column

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Entropy(PlayGolf) = ***E****(5,9)*

Entropy(PlayGolf) = ?

- Step 3: Calculate Entropy for Other Attributes After Split
- For the other four attributes, we need to calculate the entropy after each of the split.
  - E(PlayGolf, Outloook)
  - E(PlayGolf, Temperature)
  - E(PlayGolf, Humidity)
  - E(PlayGolf,Windy)

$$Entropy(\mathrm{S},\mathrm{T}) = \sum_{c \in T} P(c)E(c)$$

- Calculate **E**(PlayGolf, Outlook)

$$\boldsymbol{E}(PlayGolf, Outlook) = \boldsymbol{P}(Sunny)\boldsymbol{E}(Sunny) + \boldsymbol{P}(Overcast)\boldsymbol{E}(Overcast) + \boldsymbol{P}(Rainy)\boldsymbol{E}(Rainy)$$

*E(PlayGolf, Outlook) = **P**(Sunny) **E**(3,2) + P(Overcast) **E**(4,0) + **P**(rainy) E(2,3)*

- Frequency Table

		PlayGolf(14)		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5

$$\boldsymbol{E}(PlayGolf, Outlook) = \frac{5}{14}\boldsymbol{E}(3,2) + \frac{4}{14}\boldsymbol{E}(4,0) + \frac{5}{14}\boldsymbol{E}(2,3)$$

		PlayGolf(14)		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5

$$\boldsymbol{E(Overcast)} = E(4{,}0)$$

$$= -\left(\frac{4}{4}\log_2\frac{4}{4}\right) - \left(\frac{0}{4}\log_2\frac{0}{4}\right)$$

$$= -(0) - (0)$$

$$= \mathbf{0}$$

$$\boldsymbol{E(Sunny)} = E(3{,}2)$$

$$= -\left(\frac{3}{5}\log_2\frac{3}{5}\right) - \left(\frac{2}{5}\log_2\frac{2}{5}\right)$$

$$= -(0.60\ \log_2 0.60) - (0.40\log_2 0.40)$$

$$= -(0.60 * 0.737) - (0.40 * 0.529)$$

$$= \mathbf{0.971}$$

$$\boldsymbol{E(Rainy)} = E(2{,}3)$$

$$= -\left(\frac{2}{5}\log_2\frac{2}{5}\right) - \left(\frac{3}{5}\log_2\frac{3}{5}\right)$$

$$= -(0.40\ \log_2 0.40) - (0.6\log_2 0.60)$$

$$= \mathbf{0.971}$$

E(PlayGolf, Outlook) = ***P****(Sunny)* ***E****(3,2)* + *P(Overcast)* ***E****(4,0)* + ***P****(rainy) E(2,3)*

$$\boldsymbol{E}(PlayGolf, Outlook) = \frac{5}{14}\boldsymbol{E}(3,2) + \frac{4}{14}\boldsymbol{E}(4,0) + \frac{5}{14}\boldsymbol{E}(2,3)$$

$$\begin{aligned} E(\boldsymbol{Sunny}) &= E(3,2) \\ &= -\left(\frac{3}{5}\log_2\frac{3}{5}\right) - \left(\frac{2}{5}\log_2\frac{2}{5}\right) \\ &= -(0.60\ \log_2 0.60) - (0.40 \log_2 0.40) \\ &= -(0.60 * 0.737) - (0.40 * 0.529) \\ &= \mathbf{0.971} \end{aligned}$$

$$\begin{aligned} E(\boldsymbol{Overcast}) &= E(4,0) \\ &= -\left(\frac{4}{4}\log_2\frac{4}{4}\right) - \left(\frac{0}{4}\log_2\frac{0}{4}\right) \\ &= -(0) - (0) \\ &= \mathbf{0} \end{aligned}$$

$$\begin{aligned} E(\boldsymbol{Rainy}) &= E(2,3) \\ &= -\left(\frac{2}{5}\log_2\frac{2}{5}\right) - \left(\frac{3}{5}\log_2\frac{3}{5}\right) \\ &= -(0.40\ \log_2 0.40) - (0.6 \log_2 0.60) \\ &= \mathbf{0.971} \end{aligned}$$

$$\begin{aligned} \boldsymbol{E}(PlayGolf, Outlook) &= \frac{5}{14}\boldsymbol{E}(3,2) + \frac{4}{14}\boldsymbol{E}(4,0) + \frac{5}{14}\boldsymbol{E}(2,3) \\ &= \frac{5}{14}0.971 + \frac{4}{14}0.0 + \frac{5}{14}0.971 \\ &= 0.357 * 0.971 + 0.0 + 0.357 * 0.971 \\ &= \mathbf{0.693} \end{aligned}$$

- E(PlayGolf, Temperature)

		PlayGolf(14)		
		Yes	No	
Temperature	Hot	2	2	4
	Cold	3	1	4
	Mild	4	2	6

$$E(PlayGolf, Temperature) = P(Hot)\ E(2,2) + P(Cold)\ E(3,1) + P(Mild)\ E(4,2)$$

E (PlayGolf, Temperature) = **4/14 * E**(Hot) + **4/14 * E**(Cold) + **6/14 * E**(Mild)

E (PlayGolf, Temperature) = **4/14 * E**(2, 2) + **4/14 * E**(3, 1) + **6/14 * E**(4, 2)

E (PlayGolf, Temperature) = 4/14 * -(2/4 log 2/4) – (2/4 log 2/4)

+ 4/14 * -(3/4 log 3/4) – (1/4 log 1/4)

+ 6/14 * -(4/6 log 4/6) – (2/6 log 2/6)

E (PlayGolf, Temperature) = 5/14 * 1.0

+ 4/14 * 1.811

+ 5/14 * 0.918

= **0.911**

- E(PlayGolf, Humidity)

		PlayGolf(14)		
		Yes	No	
Humidity	High	3	4	7
	Normal	6	1	7

E (PlayGolf, Humidity) = **7/14 * E**(High) + **7/14 * E**(Normal)

E (PlayGolf, Humidity) = **7/14 * E**(3, 2) + **7/14 * E**(4, 0)

E (PlayGolf, Humidity) = 7/14 * -(3/7 log 3/7) – (4/7 log 4/7)

\+ 7/14 * -(6/7 log 6/7) – (1/7 log 1/7)

E (PlayGolf, Humidity) = 7/14 * 0.985

\+ 7/14 * 0.592

= **0.788**

- E(PlayGolf,Windy)

		PlayGolf(14)		
		Yes	No	
Windy	TRUE	3	3	6
	FALSE	6	2	8

E(PlayGolf,Windy) = ?

- E(PlayGolf, Outloook) = **0.693**
- E(PlayGolf, Temperature) = **0.911**
- E(PlayGolf, Humidity) = **0.788**
- E(PlayGolf,Windy) = ?

- Step 4: Calculating Information Gain for Each Split
  The information gain is calculated from the split using each of the attributes. Then the attribute with the largest information gain is used for the split.

$$Gain(S,T) = Entropy(S) - Entropy(S,T)$$

- For example

$$Gain(PlayGolf, Outlook) = Entropy(PlayGolf) - Entropy(PlayGolf, Outlook)$$

Gain(PlayGolf, Outlook) = Entropy(PlayGolf) – Entropy(PlayGolf, Outlook)
= 0.94 – 0.693 **= 0.247**

Gain(PlayGolf, Temperature) = Entropy(PlayGolf) – Entropy(PlayGolf, Temparature)
= 0.94 – 0.911 ***= 0.029***

Gain(PlayGolf, Humidity) = Entropy(PlayGolf) – Entropy(PlayGolf, Humidity)
= 0.94 – 0.788 ***= 0.152***

Gain(PlayGolf, Windy) = Entropy(PlayGolf) – Entropy(PlayGolf, Windy)
= ?

- Having calculated all the information gain, we now choose the attribute that gives the highest information gain after the split.

- Step 5 : Perform the First Split

- we still need to split the tree further.

- Create Subtables

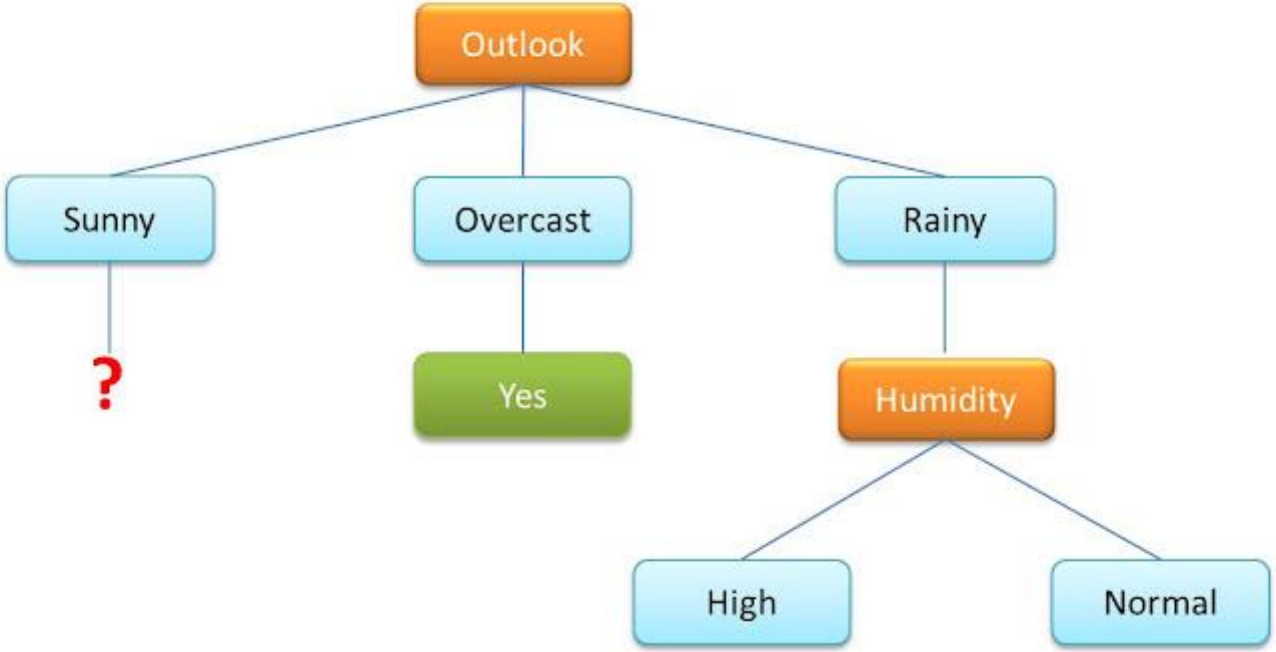
Outlook	Temperature	Humidity	Windy	Play Golf
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	High	TRUE	No

Overcast	Hot	High	FALSE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes

Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

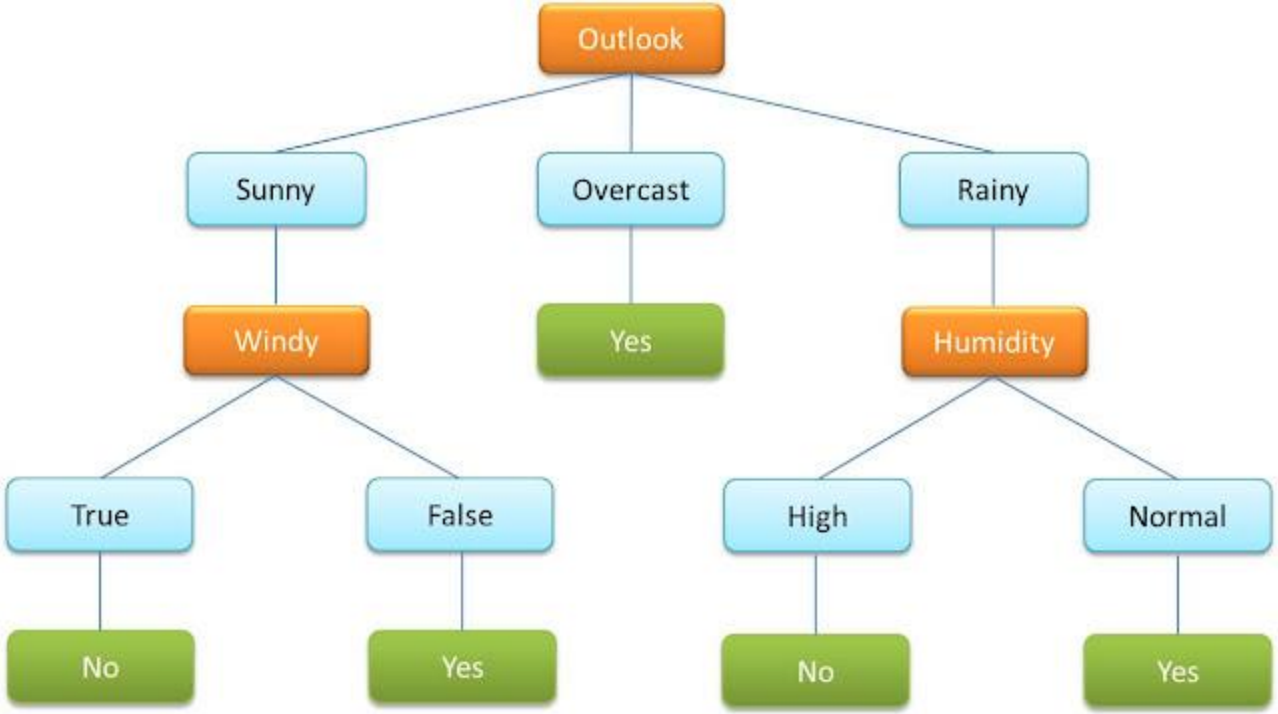
Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No

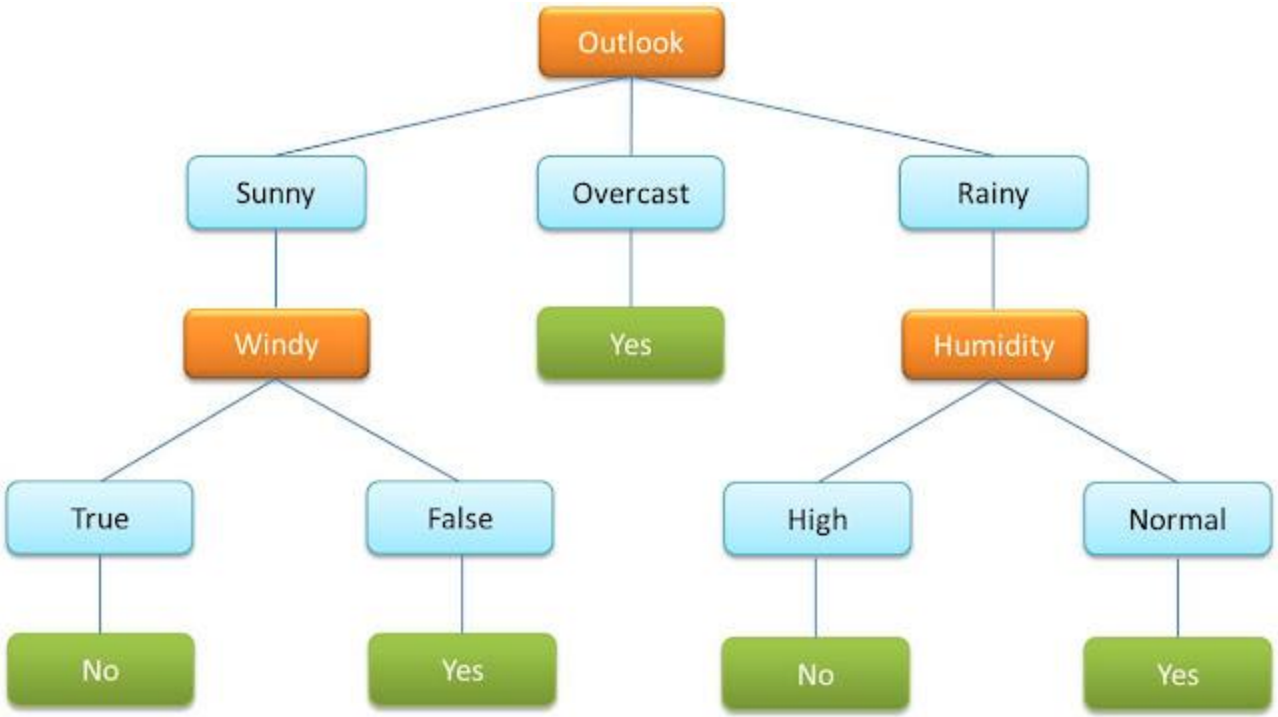
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes



Outlook	Temperature	Humidity	Windy	Play Golf
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes

Sunny	Cool	Normal	TRUE	No
Sunny	Mild	High	TRUE	No


Outlook
Sunny
Overcast
Rainy
Windy
Yes
Humidity
True
False
High
Normal
No
Yes
No
Yes




```
if(outlook.equals(“Rainy"))

{ if(humidity.equals("high")){
              class="no";
}
elseif(humidity.equals("normal")){
              class="yes";
}
}elseif(outlook.equals("overcast")){
              class="yes" }
elseif(outlook.equals(“Sunny"))
{ if(windy.equals("FALSE"))
{ class="yes"; }
elseif(windy.equals("TRUE"))
{ class="no";
}
 }
```

making predictions using new (unseen) data

- Next Lecture