

# Computational Thinking

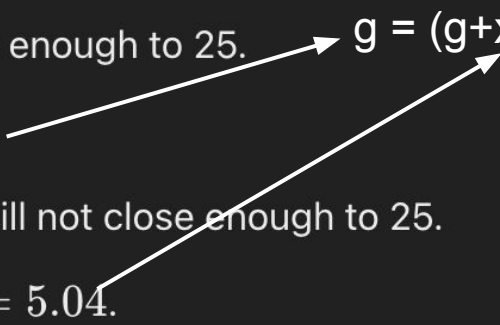
# What is Computational Thinking?

- Declarative Knowledge
  - Statements of Facts
    - “the square root of  $x$  is a number  $y$  such that  $y * y = x$ ”
- Imperative Knowledge
  - How to or recipes for deducing information
    - Heron of Alexandria method for finding the square root of a number, call it  $x$ , can be summarized as:
      - **Start with a guess:** Set initial guess,  $g$ .
      - **Check condition:** If  $g * g$  is close enough to  $x$ ,  $g$  is the answer.
      - **Update guess:** If not, update guess:  $g = (g + x/g) / 2$ .
      - **Repeat:** Continue updating until the guess is accurate.

# What is Computational Thinking?

## Example: Finding the Square Root of 25

1. **Initial Guess:** Set  $g$  to some arbitrary value, e.g., 3.
2. **Check Condition:**  $3 \times 3 = 9$ , which is not close enough to 25.
3. **Update Guess:** Set  $g = (3 + 25/3)/2 = 5.67$ .
4. **Check Again:**  $5.67 \times 5.67 = 32.15$ , which is still not close enough to 25.
5. **Update Guess:** Set  $g = (5.67 + 25/5.67)/2 = 5.04$ .
6. **Final Check:**  $5.04 \times 5.04 = 25.4$ , which is close enough to 25. We stop and declare 5.04 to be an adequate approximation to the square root of 25.

$$g = (g+x/g)/2$$


# Computational Thinking

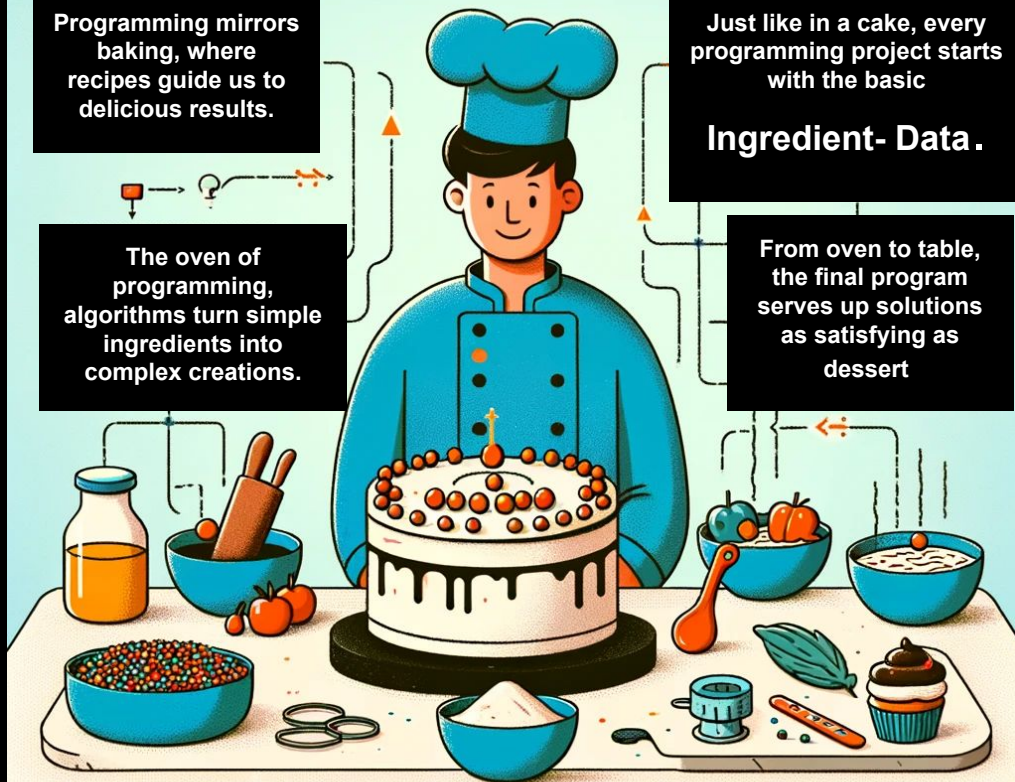
Programming mirrors baking, where recipes guide us to delicious results.

Just like in a cake, every programming project starts with the basic

**Ingredient- Data .**

The oven of programming, algorithms turn simple ingredients into complex creations.

From oven to table, the final program serves up solutions as satisfying as dessert



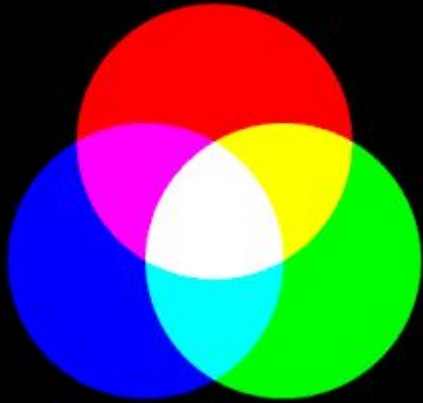
# Representation



Binary									Decimal			
MSB							LSB	=				
<b>00000000</b>								=	<b>000</b>			
	<sup>7</sup>	<sup>6</sup>	<sup>5</sup>	<sup>4</sup>	<sup>3</sup>	<sup>2</sup>	<sup>1</sup>	<sup>0</sup>		<sup>2</sup>	<sup>1</sup>	<sup>0</sup>
	<b>2</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>2</b>		<b>10</b>	<b>10</b>	<b>10</b>
	0 +	0 +	0 +	0 +	0 +	0 +	0 +	0 +	=	0 +	0 +	0

$\underbrace{(\cdot)}_{\leftarrow} < \underbrace{(\cdot)}_{\rightarrow} > \underbrace{(\cdot)}_{\leftarrow} =$   
 $\underbrace{(\cdot)}_{\leftarrow} < \underbrace{(\cdot)}_{\rightarrow} > \underbrace{(\cdot)}_{\leftarrow} =$   
 $\underbrace{(\cdot)}_{\leftarrow} < \underbrace{(\cdot)}_{\rightarrow} > \underbrace{(\cdot)}_{\leftarrow} =$   
 $\underbrace{(\cdot)}_{\leftarrow} < \underbrace{(\cdot)}_{\rightarrow} > \underbrace{(\cdot)}_{\leftarrow} =$

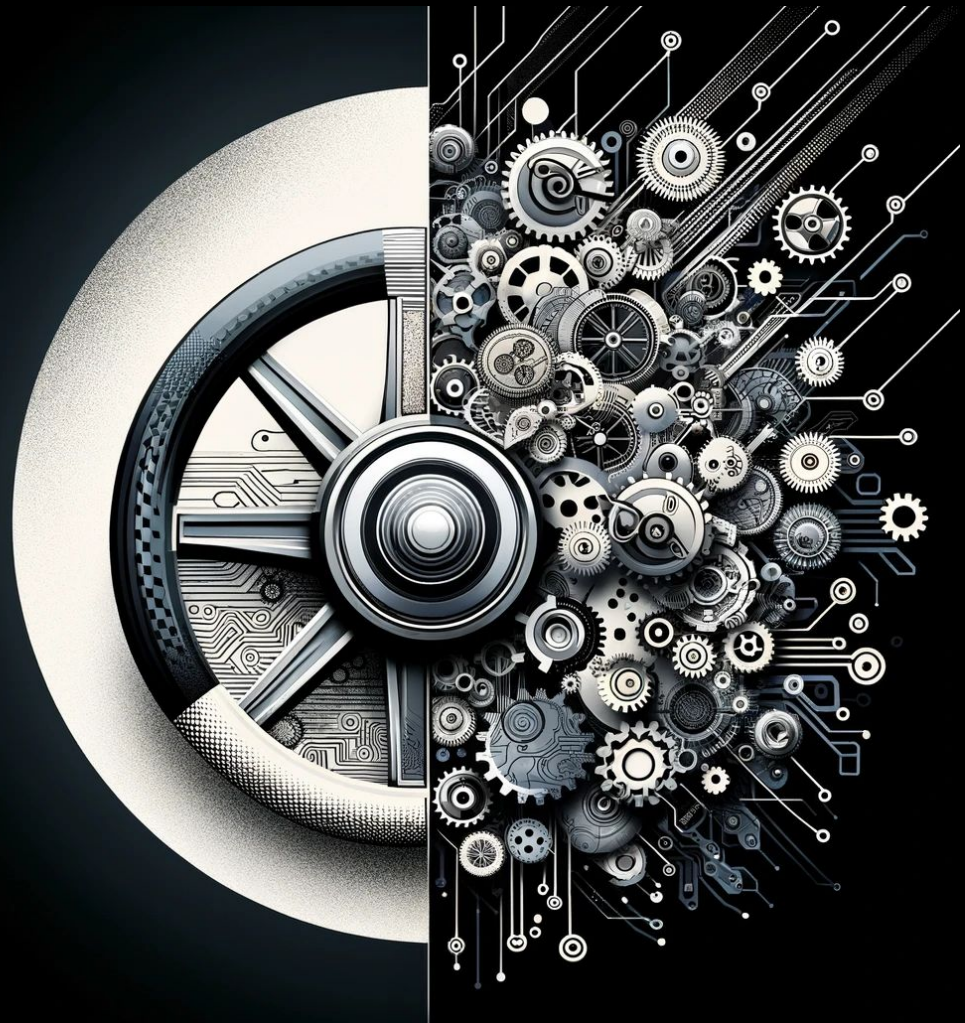
ASCII Char	Hex	Bin	ASCII Char	Hex	Bin		
65	A	41	0100 0001	97	a	61	0110 0001
66	B	42	0100 0010	98	b	62	0110 0010
67	C	43	0100 0011	99	c	63	0110 0011
68	D	44	0100 0100	100	d	64	0110 0100
69	E	45	0100 0101	101	e	65	0110 0101
70	F	46	0100 0110	102	f	66	0110 0110
71	G	47	0100 0111	103	g	67	0110 0111
72	H	48	0100 1000	104	h	68	0110 1000
73	I	49	0100 1001	105	i	69	0110 1001
74	J	4A	0100 1010	106	j	6A	0110 1010
75	K	4B	0100 1011	107	k	6B	0110 1011
76	L	4C	0100 1100	108	l	6C	0110 1100
77	M	4D	0100 1101	109	m	6D	0110 1101
78	N	4E	0100 1110	110	n	6E	0110 1110
79	O	4F	0100 1111	111	o	6F	0110 1111
80	P	50	0101 0000	112	p	70	0111 0000
81	Q	51	0101 0001	113	q	71	0111 0001
82	R	52	0101 0010	114	r	72	0111 0010
83	S	53	0101 0011	115	s	73	0111 0011
84	T	54	0101 0100	116	t	74	0111 0100
85	U	55	0101 0101	117	u	75	0111 0101
86	V	56	0101 0110	118	v	76	0111 0110
87	W	57	0101 0111	119	w	77	0111 0111
88	X	58	0101 1000	120	x	78	0111 1000
89	Y	59	0101 1001	121	y	79	0111 1001
90	Z	5A	0101 1010	122	z	7A	0111 1010



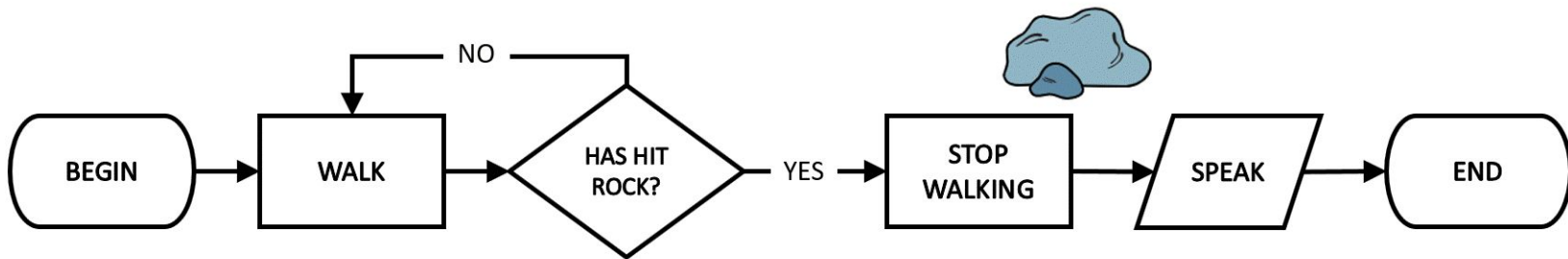
1	Black	RGB(0,0,0)
2		RGB(255,255,255)
3	Red	RGB(255,0,0)
4	Green	RGB(0,255,0)
5	Blue	RGB(0,0,255)
6	Yellow	RGB(255,255,0)
7	Magenta	RGB(255,0,255)
8	Cyan	RGB(0,255,255)
9	Brown	RGB(128,0,0)
10	Dark Green	RGB(0,128,0)
11	Dark Blue	RGB(0,0,128)
12	Olive	RGB(128,128,0)
13	Purple	RGB(128,0,128)
14	Teal	RGB(0,128,128)
15	Light Gray	RGB(192,192,192)
16	Dark Gray	RGB(128,128,128)
17	Light Purple	RGB(153,153,255)
18	Dark Purple	RGB(153,51,102)
19	Light Yellow	RGB(255,255,204)
20	Light Cyan	RGB(204,255,255)
21	Dark Purple	RGB(102,0,102)
22	Light Red	RGB(255,128,128)
23	Dark Blue	RGB(0,102,204)
24	Light Purple	RGB(204,204,255)
25	Dark Blue	RGB(0,0,128)
26	Magenta	RGB(255,0,255)
27	Yellow	RGB(255,255,0)
28	Cyan	RGB(0,255,255)

29	Purple	RGB(128,0,128)
30	Dark Red	RGB(128,0,0)
31	Dark Teal	RGB(0,128,128)
32	Dark Blue	RGB(0,0,255)
33	Cyan	RGB(0,204,255)
34	Light Cyan	RGB(204,255,255)
35	Light Green	RGB(204,255,204)
36	Light Yellow	RGB(255,255,153)
37	Light Blue	RGB(153,204,255)
38	Light Pink	RGB(255,153,204)
39	Light Purple	RGB(204,153,255)
40	Light Orange	RGB(255,204,153)
41	Dark Blue	RGB(51,102,255)
42	Teal	RGB(51,204,204)
43	Light Green	RGB(153,204,0)
44	Yellow	RGB(255,204,0)
45	Orange	RGB(255,153,0)
46	Dark Orange	RGB(255,102,0)
47	Dark Purple	RGB(102,102,153)
48	Dark Gray	RGB(150,150,150)
49	Dark Blue	RGB(0,51,102)
50	Dark Green	RGB(51,153,102)
51	Dark Green	RGB(0,51,0)
52	Dark Green	RGB(51,51,0)
53	Dark Brown	RGB(153,51,0)
54	Dark Purple	RGB(153,51,102)
55	Dark Blue	RGB(51,51,153)
56	Dark Gray	RGB(51,51,51)

**Abstraction**



# Pseudocode



# Algorithms



*"How come an algorithm can figure out what I want for my birthday,  
but you still think I want a blender?"*



# **Problem Solving**

How I dance around my  
Books ?



## **Step 1: Define the Problem**

- **Input: A mixed collection of books.**
- **Desired Output: Books organized neatly on a shelf, possibly sorted by genre, author, or title.**

## **Step 2: Gather and Analyze Data**

- **Take inventory of the books you have. Note their genres, authors, and any other relevant attributes.**

## **Step 3: Devise a Plan (Algorithm)**

- **Decide on a sorting criterion (e.g., by genre, then by author).**
- **Design a step-by-step procedure:**
  1. **Group books by genre.**
  2. **Within each genre, sort books by author.**
  3. **If there are multiple books by the same author, sort them by title.**



#### **Step 4: Execute the Plan**

**Physically sort the books according to the steps outlined in your plan.**

#### **Step 5: Review and Optimize**

**Once the books are organized, review the arrangement. Is there a more efficient way to categorize them for easier access?**

# Programming Language

# What is a Programming Language?

**Definition:** A formal set of instructions used to communicate with computers to perform specific tasks.

- **Types:**
  - **Low-Level:** Machine Language, Assembly Language.
  - **High-Level:** Python, Java, C++, JavaScript.
- **Execution:**
  - **Interpreted:** Line-by-line (e.g., Python, JavaScript).
  - **Compiled:** Entire code at once (e.g., C, C++).
- **Key Features:**
  - **Syntax & Semantics:** Rules and meanings of code.
  - **Libraries & Frameworks:** Pre-built tools to enhance functionality.

# Python Basics

[Introduction to Python](#)

# MACHINE LEARNING

AN INTRODUCTION

# MACHINE LEARNING

## Traditional Programming



## Machine Learning

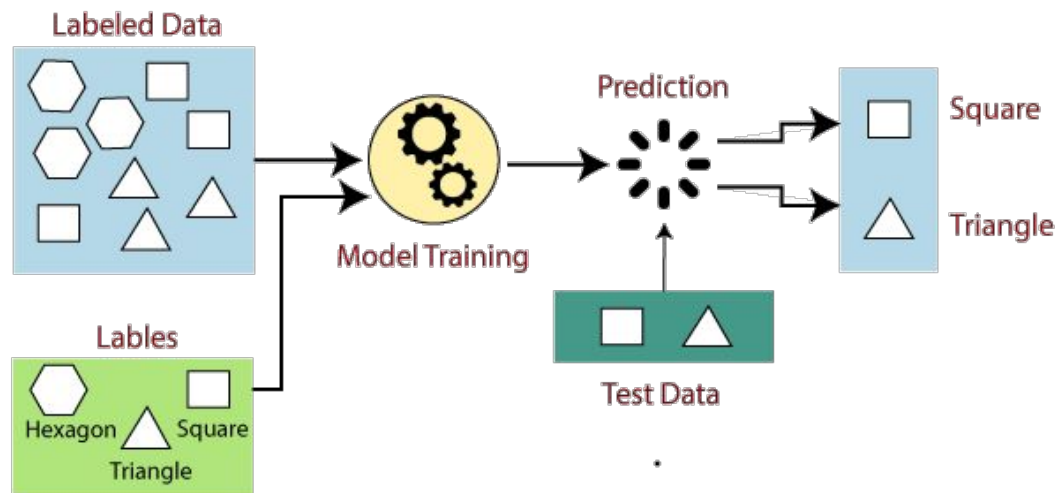


## Types:

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

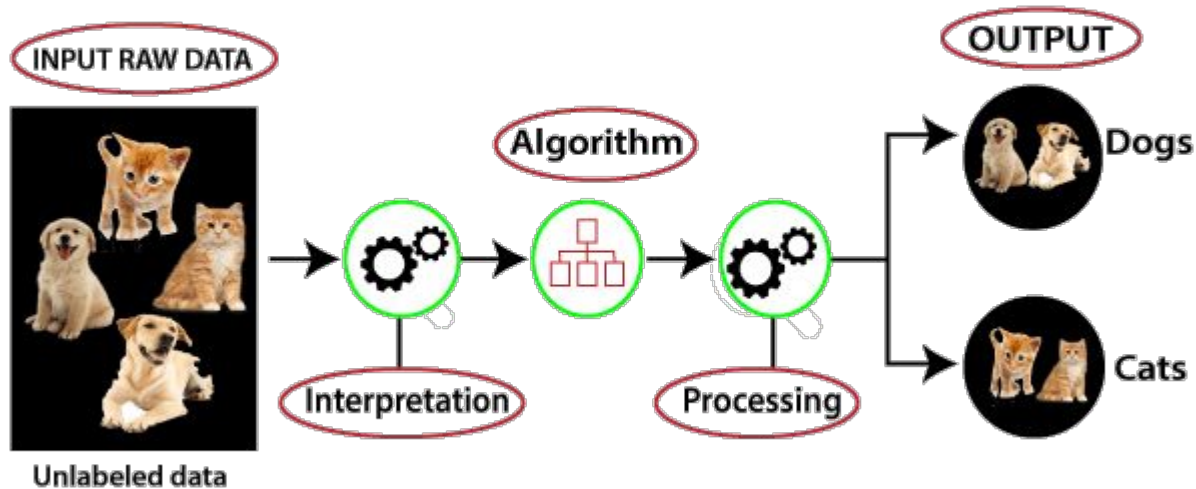
# Supervised Learning

Given: training data + desired outputs (labels)



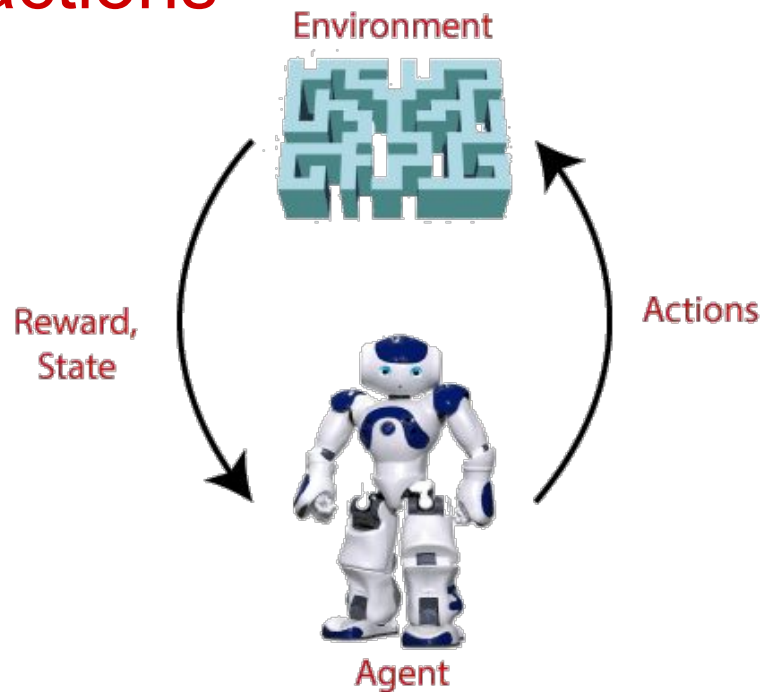
# Unsupervised Learning

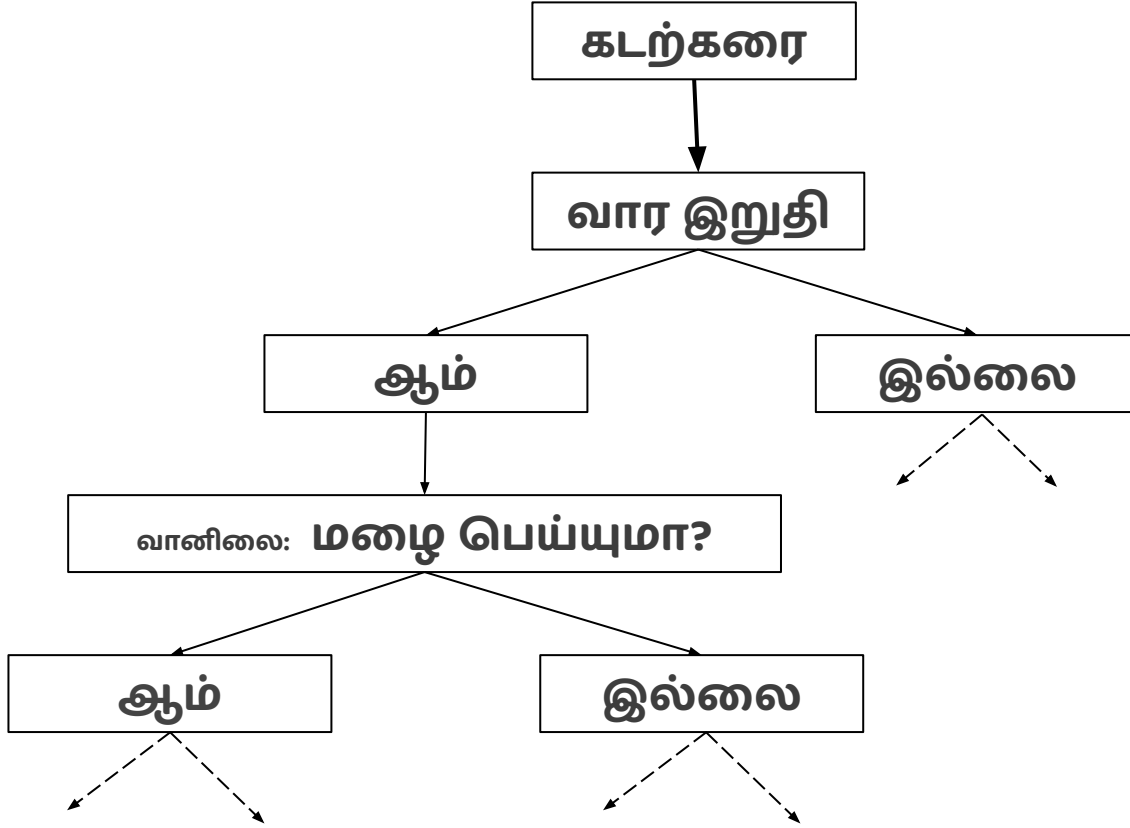
Given: training data (without desired outputs)



# Reinforcement learning

Rewards from sequence of actions





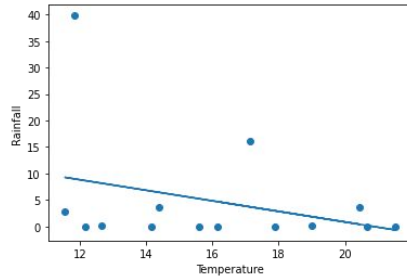
# Supervised Learning - Linear Regression

# Supervised Learning

Regression

Classification

Predicts Numeric Values



Binary



Multiclass



# Linear Regression

Independent Variable

$$Y = mx + b$$

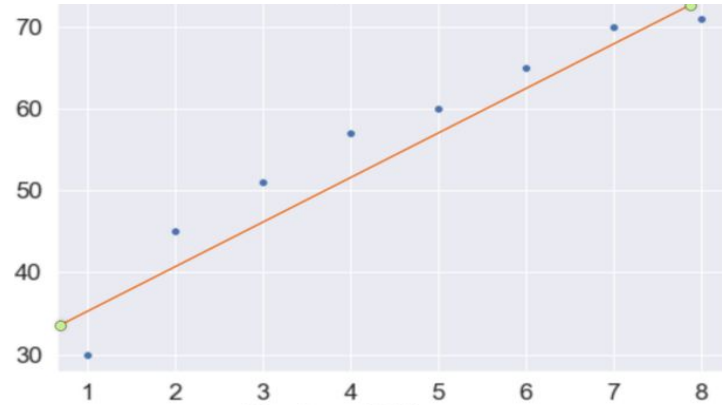
Dependent Variable

## *Short History - Linear Regression*

**Carl Friedrich Gauss - Form of Linear Algebra**

*Gauss used the least squares method to guess when and where the asteroid Ceres would appear in the night sky.*

No. of Butterflies (x)	Percent of Hunts (y)
1	30
2	45
3	51
4	57
5	60
6	65
7	70
8	71
36	449



$$Y = mx + b$$

$$m = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}$$

$$b = \frac{\sum y - m(\sum x)}{n}$$

No. of Butterflies (x)	Percent of Hunts (y)	xy	x <sup>2</sup>	y <sup>2</sup>
1	30	30	1	900
2	45	90	4	2025
3	51	153	9	2601
4	57	228	16	3249
5	60	300	25	3600
6	65	390	36	4225
7	70	490	49	4900
8	71	568	64	5041
36	449	2249	204	26541

$$m = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2} \quad b = \frac{\sum y - m(\sum x)}{n}$$

$$m = \frac{8(2249) - (36)(449)}{8(204) - (36)^2} \quad b = \frac{449 - 5.4405(36)}{8}$$

$$m = 5.4405 \quad b = 31.6429$$

$$y = mx + b$$

$$y = 5.4405x + 31.6429$$

**Hours Studied**

**Features (Input)**

**Final Exam Score**

**Target (Output)**

Hours Studied	Final Exam Score
2	65
4	75
6	85
8	95

**Linear Equations**

**Model**

**m (slope) and b (y-intercept)**

**Parameters**

**The process of finding the best m and b**

**Training**

**Using the model to estimate scores for new students**

**Prediction**

# Supervised Learning: Teaching Machines with Labeled Data

## Definition:

- Learning from labeled training data
- Goal: Predict output for new, unseen inputs

## Key Components:

- Input features ( $X$ )
- Target variable ( $Y$ )
- Mapping function:  $f(X) = Y$

## Types:

- Classification: Predicting categories
- Regression: Predicting continuous values

# Unsupervised Learning: Discovering Hidden Patterns in Data

## Definition:

- Learning from unlabeled data
- Goal: Find inherent structures in data

## Key Characteristics:

- No predefined output variable
- Explores data to find hidden patterns
- Often used for exploratory data analysis

## Main Types:

- Clustering: Grouping similar data points
- Dimensionality Reduction: Compressing data while preserving information
- Anomaly Detection: Identifying unusual data points

# Reinforcement Learning: Learning through Interaction

## Definition:

- Learning to make decisions by interacting with an environment
- Goal: Maximize cumulative reward over time

## Key Components:

- Agent: The learner or decision-maker
- Environment: The world the agent interacts with
- State: Current situation of the agent
- Action: A move the agent can make
- Reward: Feedback from the environment