ARTIFICIAL INTELLIGENCE

DAY 1

Prepared by TARSOFT SDN BHD

TABLE OF CONTENTS

Knowledge Test

Introduction to AI & Ollama Basics

Objectives

01 Participants understand the fundamentals of AI, LLMs, and Ollama, and can run an AI model locally.

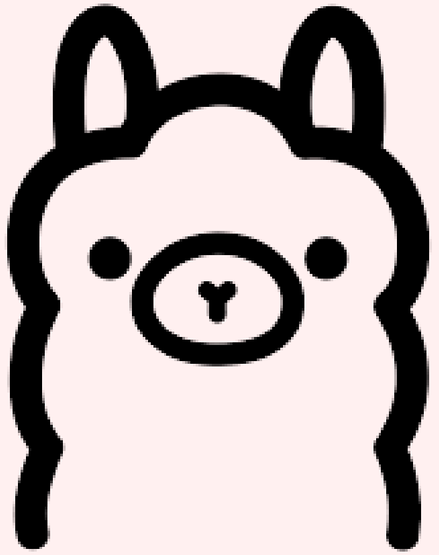
Course Introduction & Objectives

01 Understand what AI is.

02 Learn about history & evolution.

03 Explore LLMs.

04 Hands-on with Ollama.



Introduction to AI (Artificial Intelligence)

What is AI? (The Big Picture)

- Computer systems that can perform tasks that normally require human intelligence.
- Machines can learn, reason, and decide.
- Analogy: Like teaching a child new skills → but here we “teach” a computer.

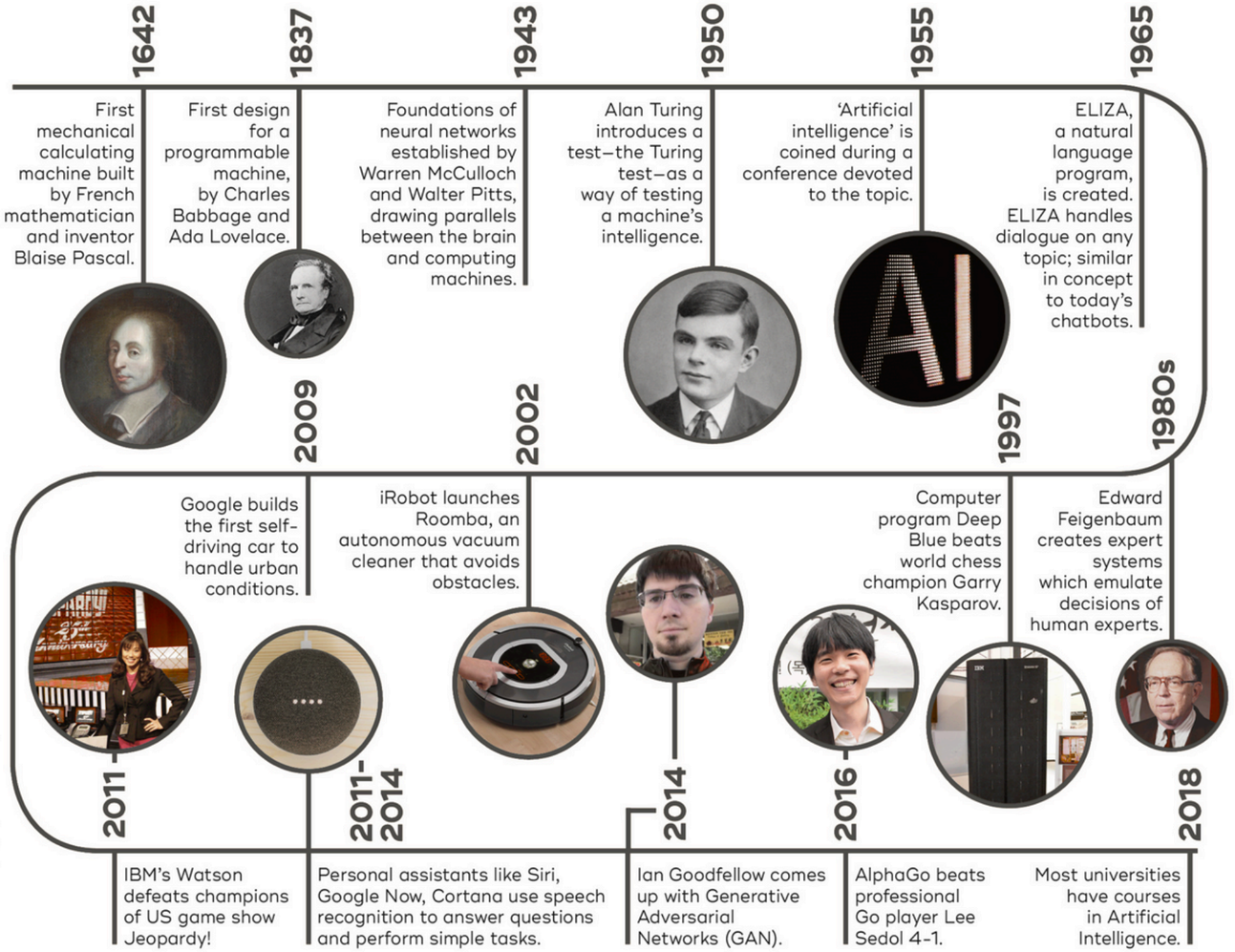
Key abilities:

- Perception (see/listen).
- Reasoning (problem-solving).
- Learning (from data).
- Language (understand & respond).

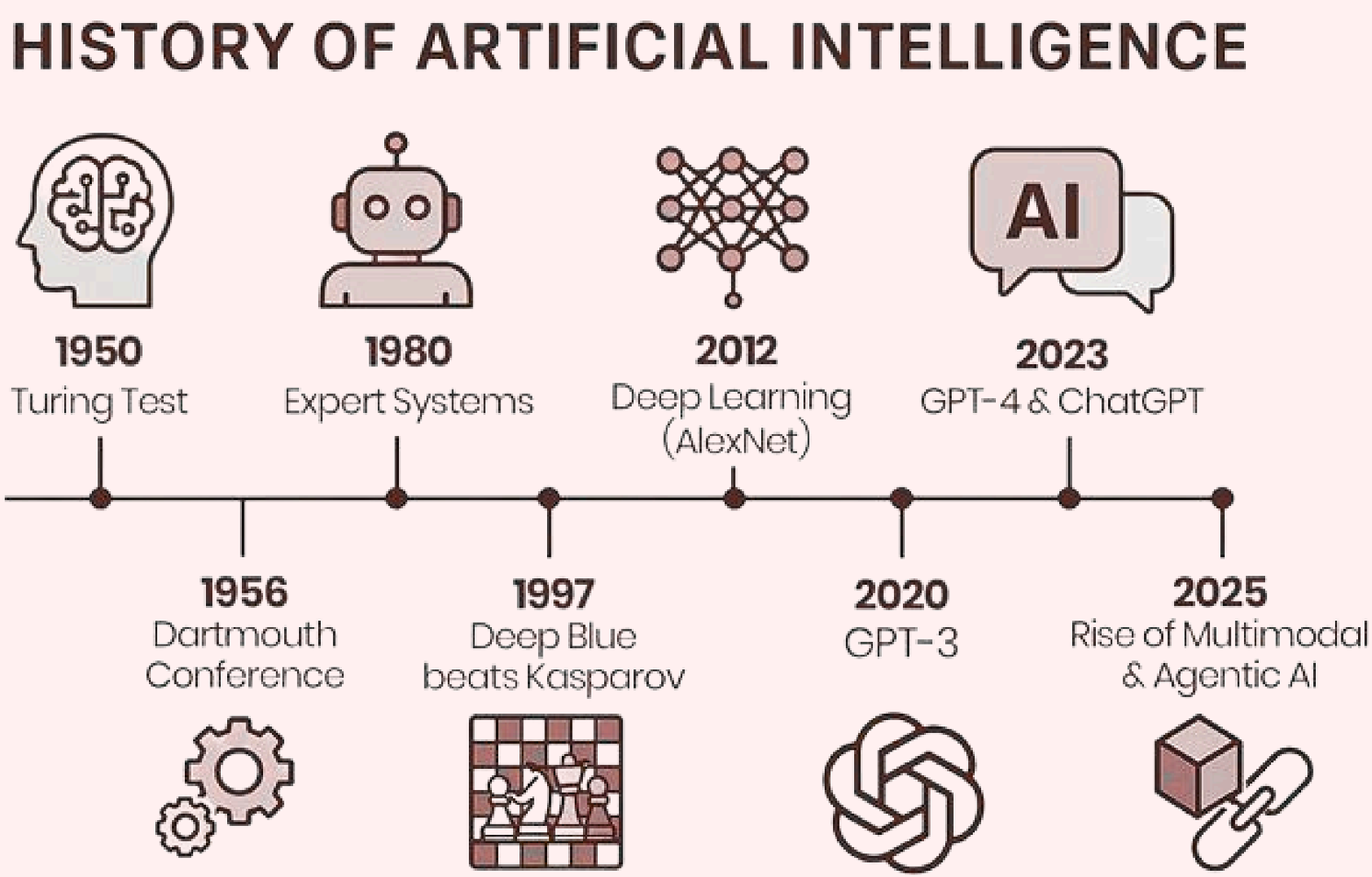
The Difference: AI vs Normal Software

Traiditional Software	AI Software
Follows fixed rules (if A → do B)	Adapts, improves, learns patterns
Calculator → only performs exact formula	Can generate creative text, like writing a poem

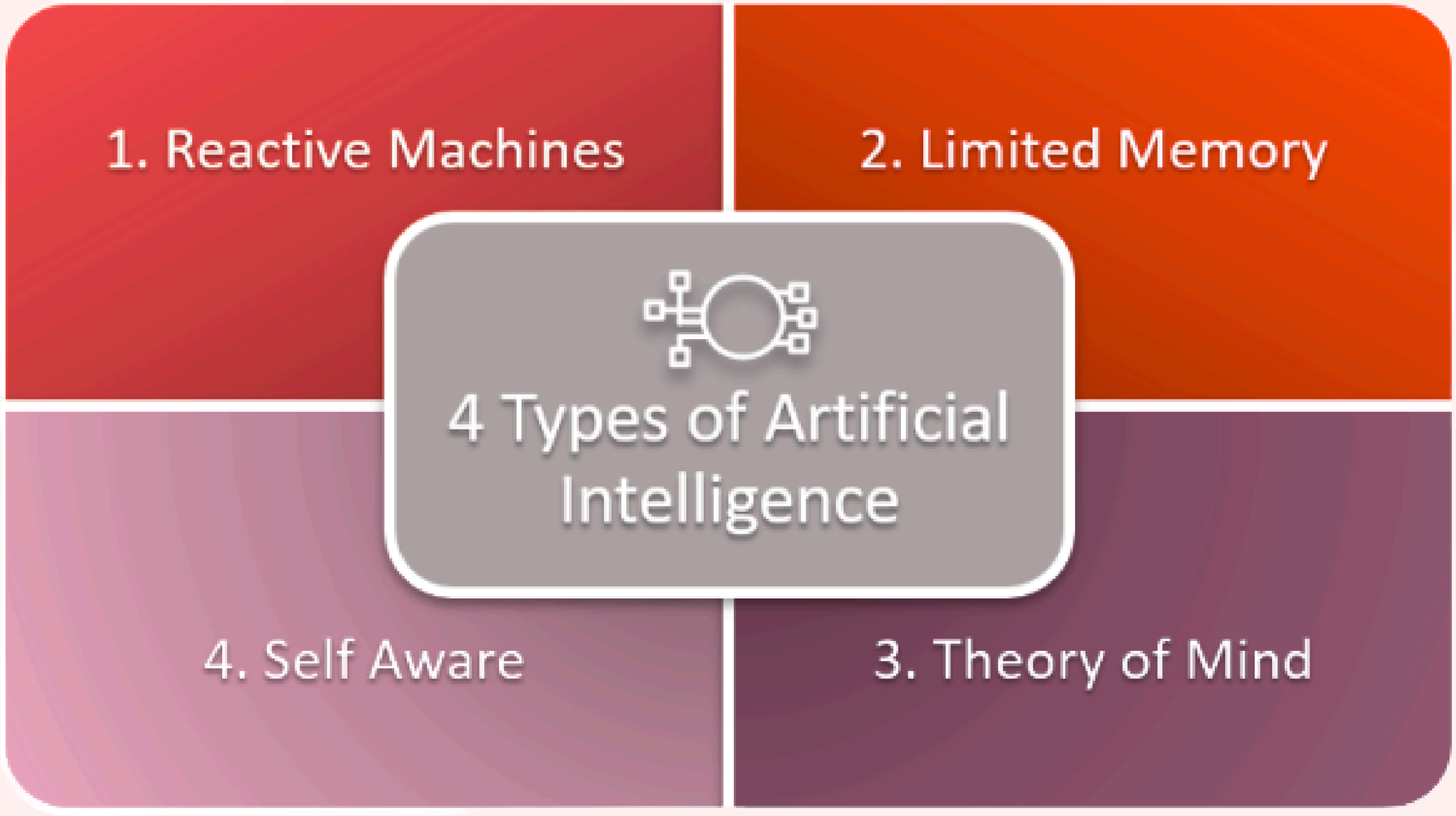
History of AI



The Difference: AI vs Normal Software



Types of AI


1. Reactive Machines
2. Limited Memory
4 Types of Artificial Intelligence
4. Self Aware
3. Theory of Mind

Types of AI

Reactive Machines

No memory, just react.

Example:

- IBM Deep Blue (chess).
- Netflix Rec Engine

Limited Memory

Learns from past data.
Reinforce Learning

Example:
Self-driving cars.

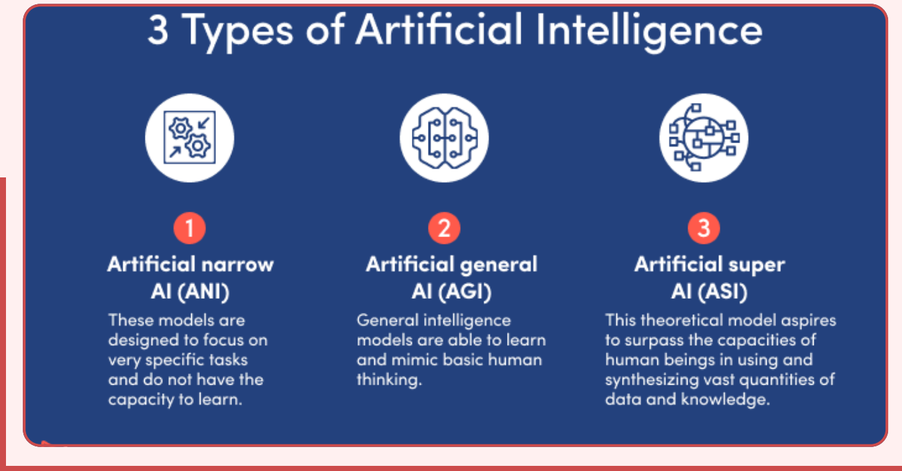
Self-Aware AI (hypothetical)

AI with consciousness.

Theory of Mind (future research)

AI that understands human emotions & intentions.

Types of AI
(by capability)



1

Specialized in one task.
Example: Siri, Google Translate.

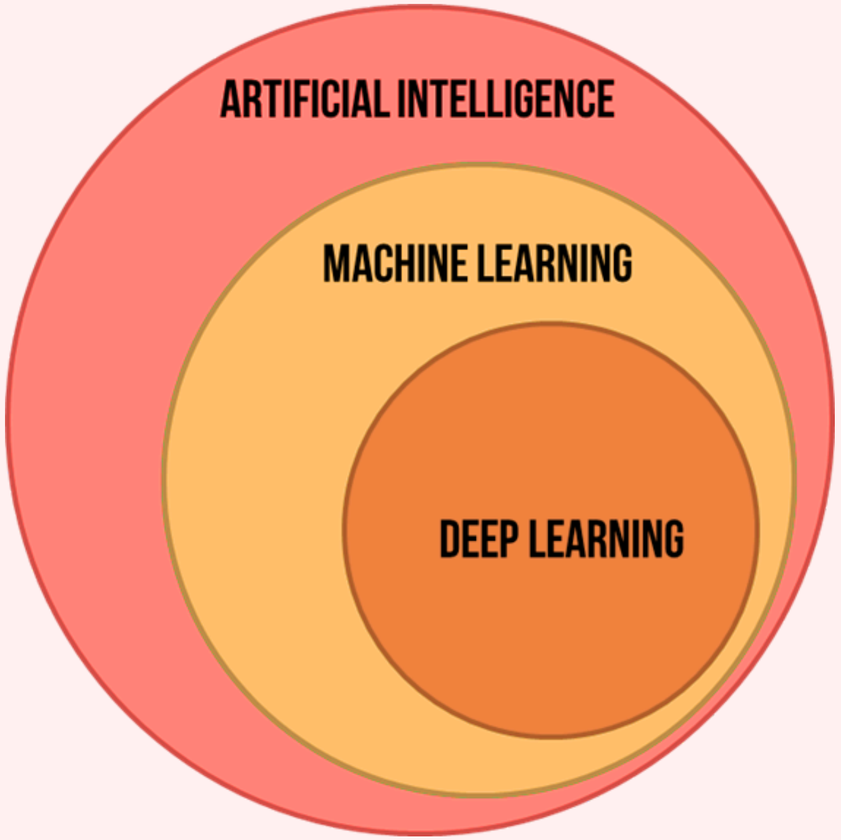
2

Human-level intelligence across all tasks.
Still a research goal, not achieved yet.

3

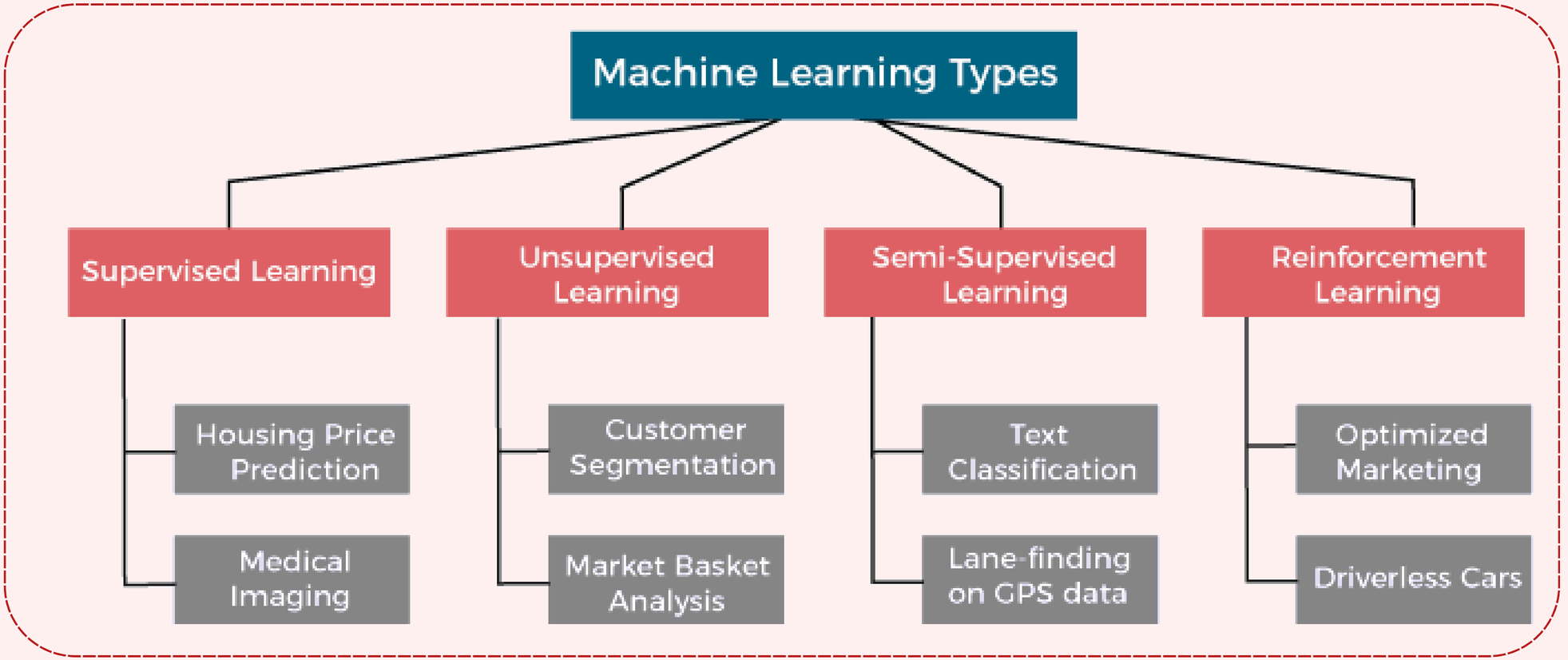
Hypothetical, beyond human intelligence.
Often seen in sci-fi.

AI vs ML vs DL



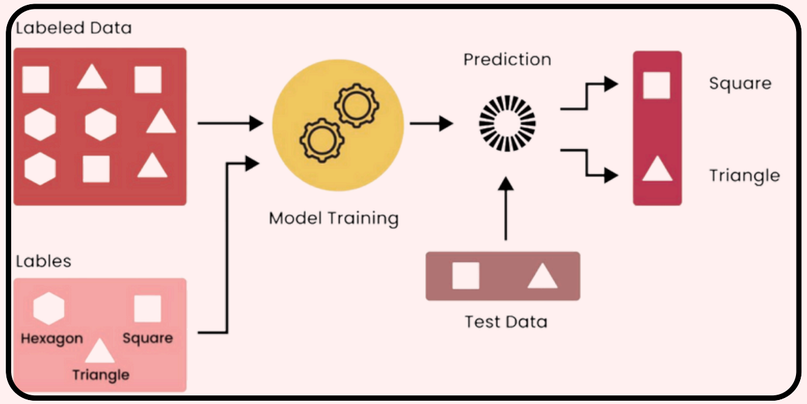
- AI is the ability of a computer to do tasks that are usually done by humans
- ML is one of the methods to “achieve” AI
- DL is a method in ML with the use of Neural Networks

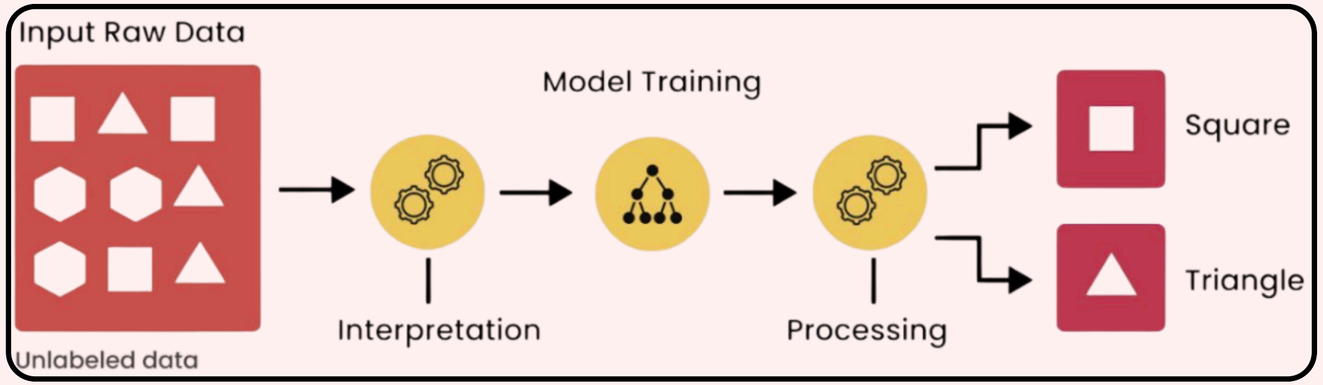
Learning types in machine learning



Supervised Learning vs Unsupervised Learning

Supervised Learning	Unsupervised Learning
• Uses labeled data to train model • It can be categorized into Classification or regression	Analyzes unlabeled data without explicit correct labels and identifies internal patterns, clusters, or hidden factors that may be present in the data.





Supervised Learning vs Unsupervised Learning

Supervised Learning models

- Naive Bayes Classifier
- Support Vector Machine(SVM)
- Linear Regression Models
- Logistic Regression
- Decision Trees
- Random Forest
- K-nearest (KNNs)

Unsupervised Learning Models

- K-means clustering
- Principal Component Analysis (PCA)

Where Do LLMs Fall in AI Learning?

LLMs are not trained in just one way.

They go through 3 main stages:

1. Pretraining – learn language patterns from huge amounts of text.
2. Fine-tuning – teach the model to follow instructions or do specific tasks.
3. Alignment (RLHF) – make sure the model's answers are safe and useful for humans.

Think of it like:

- Pretraining = going to school and reading every book.
- Fine-tuning = taking a specialized course (e.g., medicine, law).
- Alignment = a teacher correcting your behavior so you act politely and responsibly.

Pretraining (Self-Supervised Learning)

What happens: The model reads billions of sentences and learns to predict the next word.

Why it's "self-supervised": The data itself gives the answers (the "next word"), so no human labels are needed.

What it learns:

- Grammar and sentence structure.
- General knowledge (facts, reasoning patterns).
- Common sense from real-world text.

Limitation: The model only knows what's in its training data and may make things up (hallucinations).

Sentence: "The cat sat on the ___"

- Model learns the most likely word is "mat."
- By repeating this billions of times, it learns how language works.
- This is how the model becomes capable of writing essays, answering questions, or summarizing text.

Fine-Tuning (Supervised Learning)

After pretraining, the model is very smart but not very helpful.

Fine-tuning means giving it examples of questions and the best answers.

Example:

- Input: “What is the capital of France?”
- Desired Output: “Paris.”

By learning from these pairs, the model becomes better at following instructions and producing accurate answers.

This is called **supervised learning** because humans provide the “correct answers.”

Alignment with Human Preferences (RLHF)

Even after fine-tuning, the model can still give **rude, biased, or unsafe** answers.

RLHF = Reinforcement Learning with Human Feedback.

How it works:

- The model generates several possible answers
- Humans rank the answers (best → worst).
- A reward system is trained from these rankings.
- The model is adjusted to give answers humans prefer.

This makes the AI safer, more polite, and more aligned with what people expect.

AI in Daily Life

Recommender Systems

NETFLIX

YouTube

Autonomous Driving

TESLA

Healthcare

- Disease prediction
- Drug discovery

Finance

- Fraud detection
- Robo-advisors

Large Language Models (LLM)

Definition

A Large Language Model (LLM) is an AI system trained on massive amounts of text data (books, articles, websites, code, etc.) to understand and generate human-like language.
"Large" → means billions of parameters (the “neurons” inside the model).

"Language" → text-based communication (English, Malay, Chinese, code).

"Model" → mathematical system that predicts the next word in a sentence.

How it Works

Training: LLM reads billions of sentences and learns patterns of words.

Prediction: When you give it a prompt, it predicts the most likely next word repeatedly until it forms a complete answer.

Analogy:

Like a person who has read millions of books → can guess what comes next in a sentence.

Example: If you start “Once upon a…”, LLM predicts “time”.

Large Language Models (LLM)

Example of LLMs

- OpenAI GPT series (GPT-3.5, GPT-4, GPT-4o)
- Meta's LLaMA 3 (open-source, runs on Ollama)
- Mistral (lightweight & fast)
- Gemma (Google's open-source model)

What LLMs can do

- Answer questions (like Google Search, but conversational).
- Summarize long documents into key points.
- Translate between languages (English ↔ Malay).
- Generate content (emails, articles, marketing copy).
- Act as coding assistants (Python, JavaScript, SQL).

AI Ecosystem in Applications

Use Cases

- Customer Service & Chatbots
- Healthcare & Diagnostics
- Document Processing & Summarization
- Content Creation & Localization
- Transport & Mobility
- Education & Learning

Local Example

- eKYC for IC verification, government chatbots
- Company Chatbot
- Invoice Processing

AI Ecosystem in Applications

LLLM Limitations

- Hallucinations: makes up facts.
- Knowledge cutoff: cannot access real-time internet (unless connected to tools).
- Computational cost: large models need high resources.
- Bias: trained on internet text, so may reflect bias.

Ollama: What is it?

Framework / tool

Local

Large language models (LLMs)

Why Ollama?

Privacy: Data never leaves your machine.

Cost Saving: No API usage fees.

Control: Choose models, customize them.

Offline: Works even without internet.

Ollama

LM Studio

Introduction to Ollama

Advantages

- No need to send data to the cloud.
- Safer data handling.
- Less cost

Disadvantages

- Hard to set up
- Maintenance problem
- Learning curve

System prerequisites

RAM Memory	• 16 GB RAM or more → works well with moderate models (e.g., gemma:7b, mistral:7b). • 8 GB RAM → possible for smaller quantized models (:2b or :3b), but performance may lag. • 32 GB+ RAM → ideal for running multiple or larger models (llama3:70b).
GPU	• RTX3090 • H100
Disk Storage (SSD)	Model size examples: • gemma:2b → ~1.5 GB • mistral:7b → ~4 GB • llama3:70b → 40–70 GB+
Supported OS	• MacOS (Monterey or later), • Windows 10 or later, • Linux (various distros).

Installing & Configuring Ollama

- **Installer (Windows/Mac/Linux):** https://ollama.com/download

- **CLI Install:** curl -fsSL https://ollama.ai/install.sh | sh

Large Language Models (LLM)

Model	Size (approx)	Best for	Why Recommended	Suggested Specs
Mistral 7B (mistral)	~4–5 GB	General Q&A, summarization, coding help	Balanced between speed & quality	16GB RAM, quad-core CPU, SSD; GPU optional
Gemma 2B / 7B (gemma:2b, gemma:7b)	2B: ~1.5 GB 7B: ~5 GB	Lightweight chatbot, text generation	Google-optimized, smooth on laptops	8 GB RAM (2B) / 16GB RAM (7B)
LLaMA 3 8B (llama3:8b)	~8–9 GB	Deeper reasoning, knowledge tasks	Stronger reasoning than 7B models	16 GB RAM minimum , GPU helps a lot
Phi-3 Mini (3.8B) (phi3:3.8b)	~2–3 GB	Educational Q&A, smaller apps, code	Small but powerful Microsoft model	8–12 GB RAM, runs well on laptops
GPT-OSS 20B (Honorable Mention) (gpt-oss:20b)	~16–20 GB	Complex reasoning, long conversations, document tasks	Higher coherence & capability, but heavy	32 GB RAM minimum , GPU strongly recommended (RTX 3090/4090, A100), Not practical for 16 GB RAM laptops

Ollama Commands

Command	Description	Example	Notes
ollama --version	Check installed Ollama version	ollama --version	Confirms installation success
ollama help	Show list of available commands	ollama help	Good first step if stuck
ollama run <model>	Download (if not installed) and start a model	ollama run mistral	Interactive mode – type prompts directly
ollama pull <model>	Download a model without running it	ollama pull gemma:2b	Useful for pre-downloading before training
ollama list	Show all locally installed models	ollama list	Helps check what's available

Exploring Models in Ollama

Command	Description	Example	Notes
ollama show <model>	Display model details (size, parameters, etc.)	ollama show mistral	Great for explaining model differences
ollama rm <model>	Remove a model from local storage	ollama rm gemma:2b	Frees disk space
ollama serve	Start Ollama in API server mode	ollama serve	Needed for integration with apps (Python, JS, etc.)
ollama create <name> -f <Modelfile>	Build a custom model from a Modelfile	ollama create mymodel -f Modelfile	Advanced use (fine-tuning / importing GGUF)

Ollama Architecture

Core Components

- CLI (ollama run) : interact directly.
- Server Mode (ollama serve) : expose API endpoints.
- Model Files (.bin, quantized weights).

Workflow

- User query → Ollama → Model → Response returned.

(Insert diagram: User → Ollama CLI/Server → LLM → Output)

Hands-On Day 1

(Play around with models)

1

Task: run a simple prompt (“Hello, how are you?”).

2

Compare model outputs.

3

Discussion: which model fits which use case.

Q&A & Day 1 Recap

Recap: AI, LLM, Ollama, RAG.

What’s the difference between AI and Google Search?

Thank You

Any Enquiries?

tarsoft.com.my