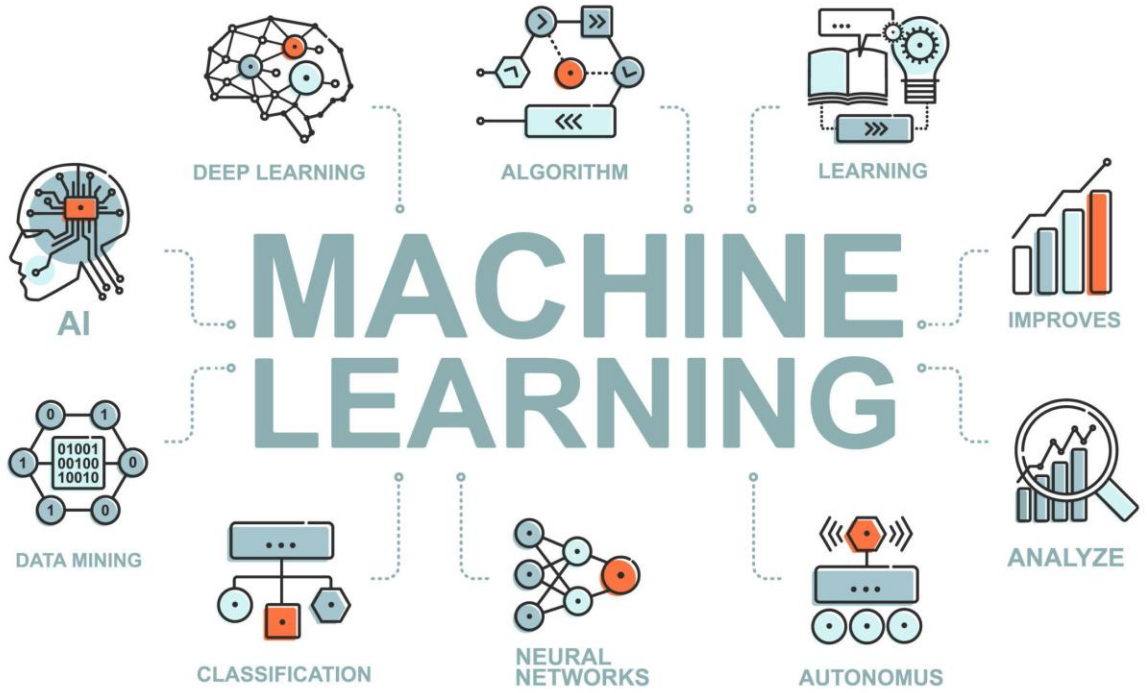
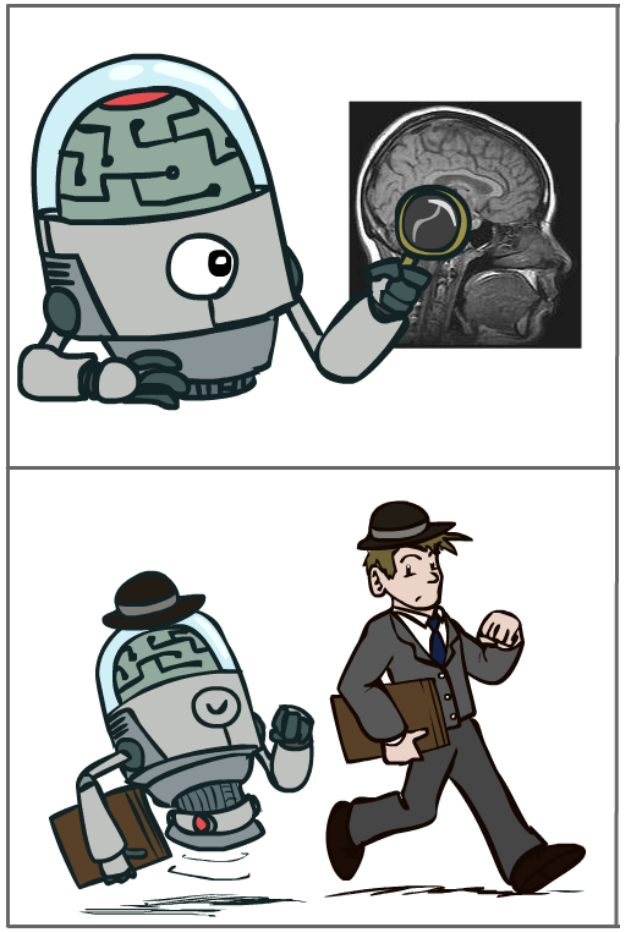
Introduction to Machine Learning



What is AI?

The science of making machines that:



emulate human behavior, enabling them to **learn**, **make decisions**, **recognize patterns**, and **solve complex problems** in a manner akin to human intelligence

AI vs. ML

- **Artificial Intelligence (AI):**
  - AI is a broad field of computer science that aims to create systems that can perform tasks that typically require human intelligence
  - It encompasses a wide range of techniques, approaches, and applications, including problem-solving, natural language understanding, speech recognition, computer vision, and decision-making
- **Machine Learning (ML):**
  - ML is a subset of AI that focuses on developing algorithms and models that enable machines to learn patterns and make predictions or decisions based on data.
  - It involves the use of algorithms to allow machines to improve their performance on a specific task over time without being explicitly programmed.

AI vs. ML vs. DL vs. Generative AI

Artificial Intelligence

AI involves techniques that equip computers to emulate human behavior, enabling them to learn, make decisions, recognize patterns, and solve complex problems in a manner akin to human intelligence.

Machine Learning

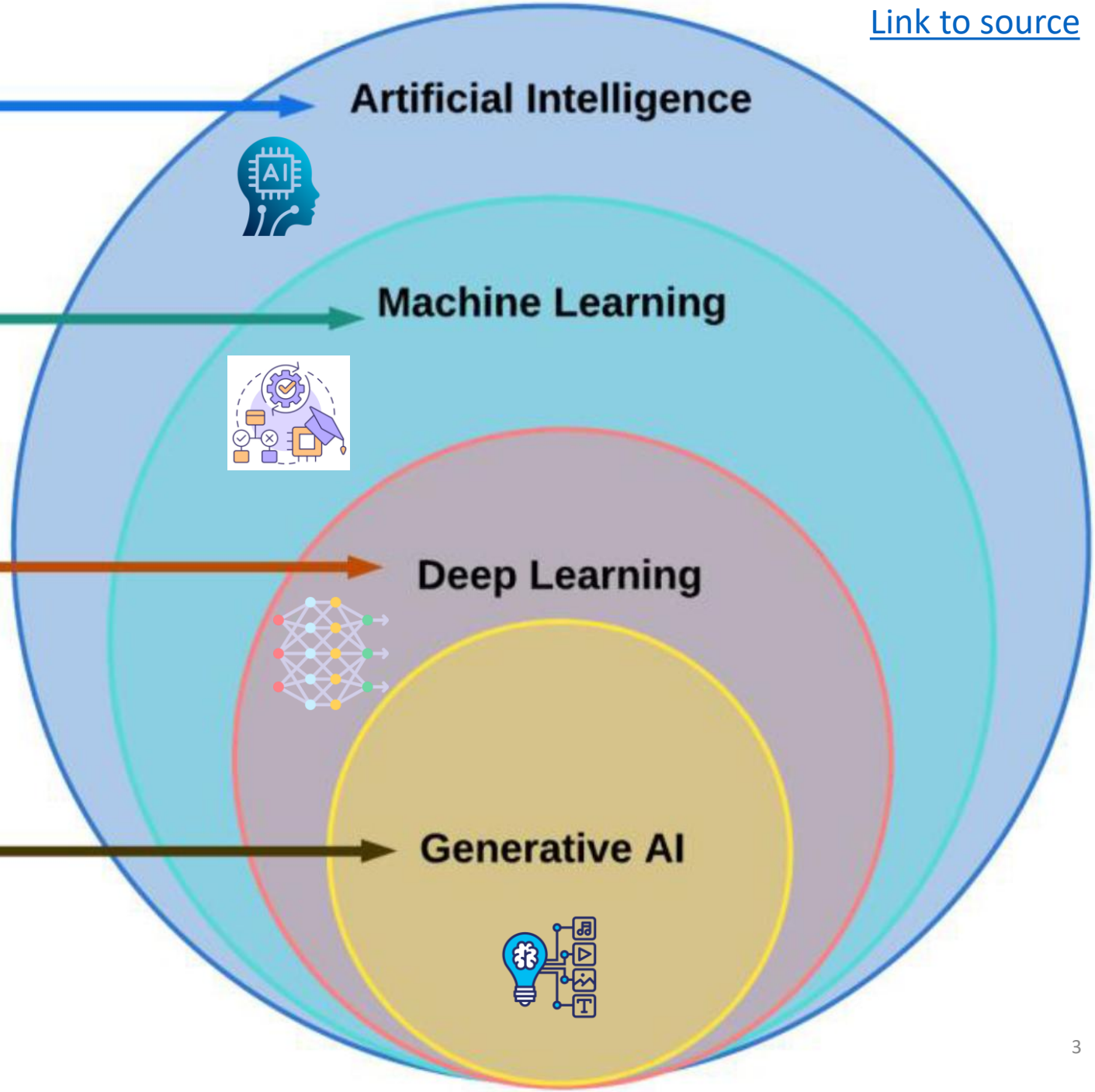
ML is a subset of AI, uses advanced algorithms to detect patterns in large data sets, allowing machines to learn and adapt. ML algorithms use supervised or unsupervised learning methods.

Deep Learning

DL is a subset of ML which uses neural networks for in-depth data processing and analytical tasks. DL leverages multiple layers of artificial neural networks to extract high-level features from raw input data, simulating the way human brains perceive and understand the world.

Generative AI

Generative AI is a subset of DL models that generates content like text, images, or code based on provided input. Trained on vast data sets, these models detect patterns and create outputs without explicit instruction, using a mix of supervised and unsupervised learning.



Link to source

Machine Learning

- **Machine learning** is the subfield of computer science that gives "computers the ability to learn without being explicitly programmed."

  - term coined by Arthur Samuel 1959 while at IBM

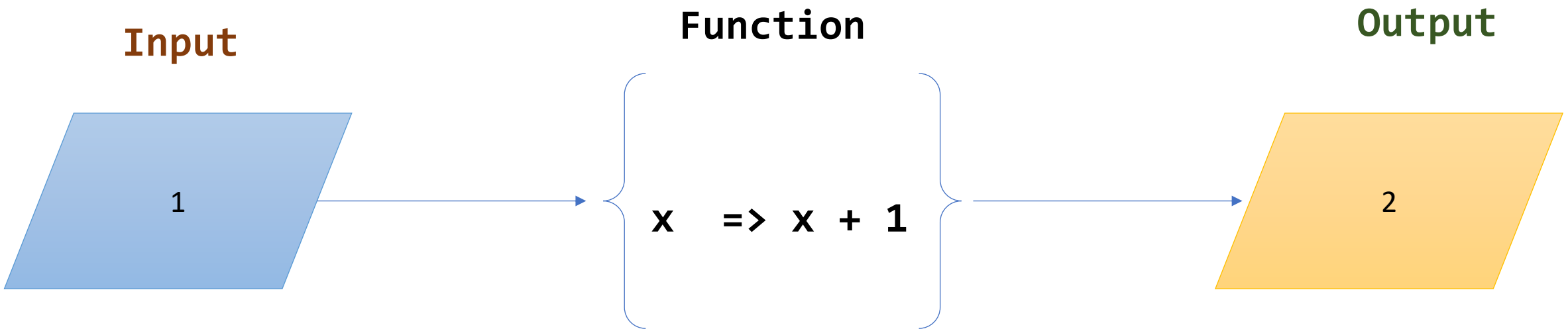
- The study of algorithms that can learn from data.

- Machine learning **focuses on the development of computer programs that can access data and use it learn for themselves**

  - expert.ai

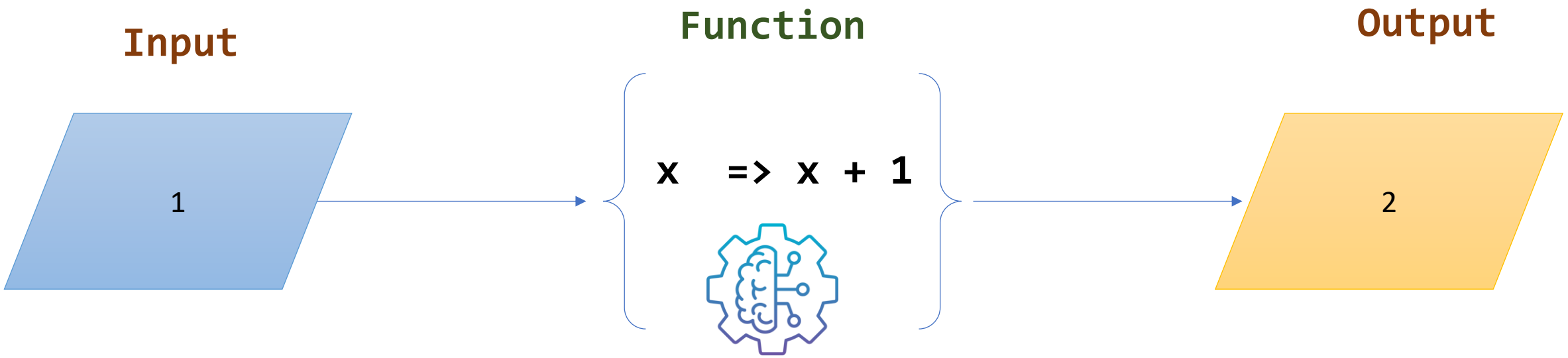
Coding : Input + Function → Output

- In conventional coding, the programming collects the **Input** and write the **Function(s)** to produce the desired **Output**



ML : Input + Output → Function

- ML is all about intelligent guessing/approximation of a function to produce the known **Output** from a given **Input**



Supervised Learning Examples:

Estimate the function f from input/output examples

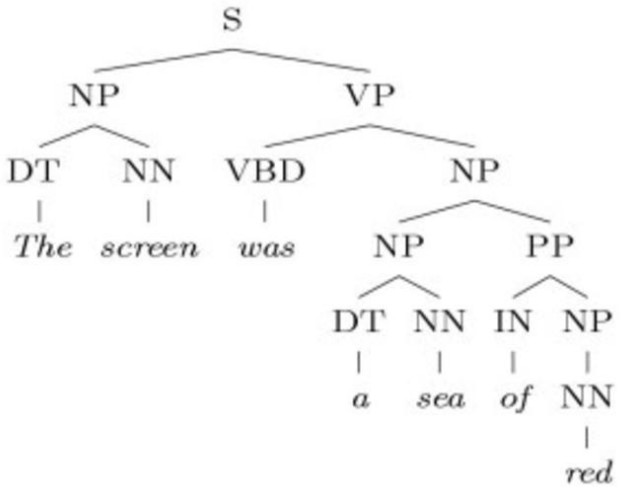
$f($  $) =$ cat

Classification

$f($  $) =$ 

Facial Detection

$f($ The screen was a sea of red $) =$



Language Parsing

Example ML Function

- Apply a prediction function to a feature representation of the image to get the desired output:

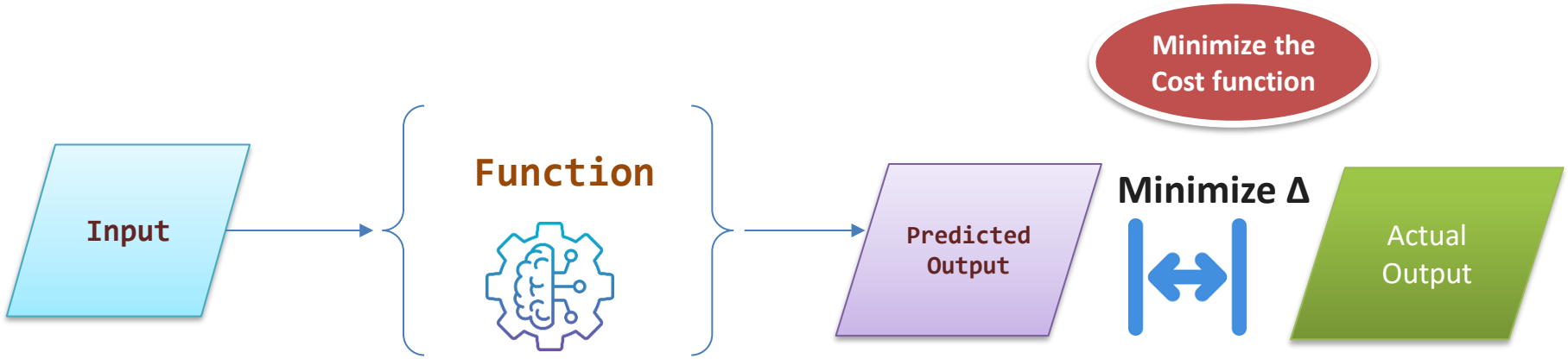
f() = “apple”

f() = “tomato”

f() = “cow”

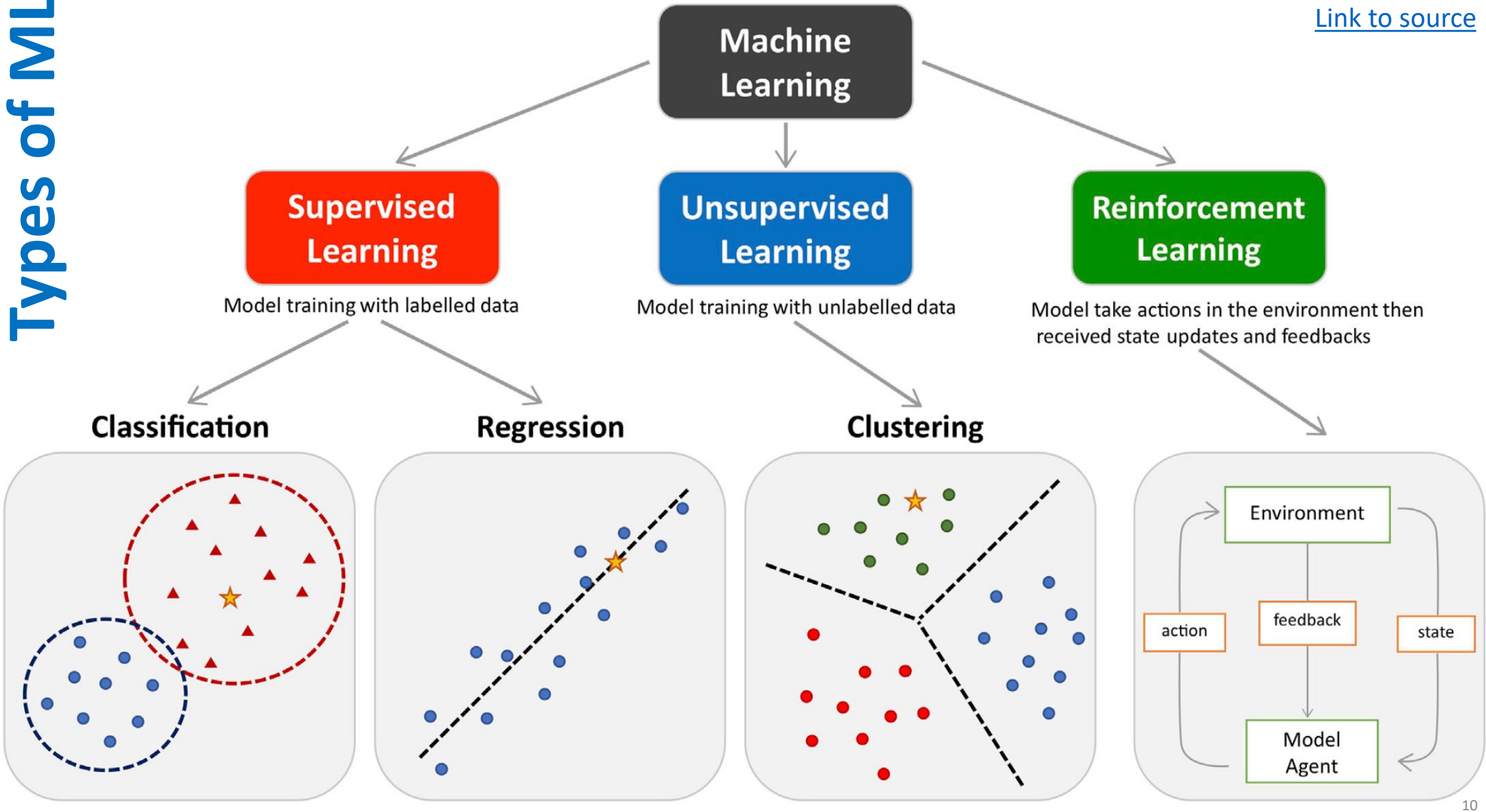
ML: learn a Function that minimizes the cost

- Start with random function parameters
- Repeat intelligent guessing/approximation of the Function parameters such that the difference between the Predicted Output the Actual Output is reduced
  - i.e., minimize a Cost function a.k.a loss, or error function



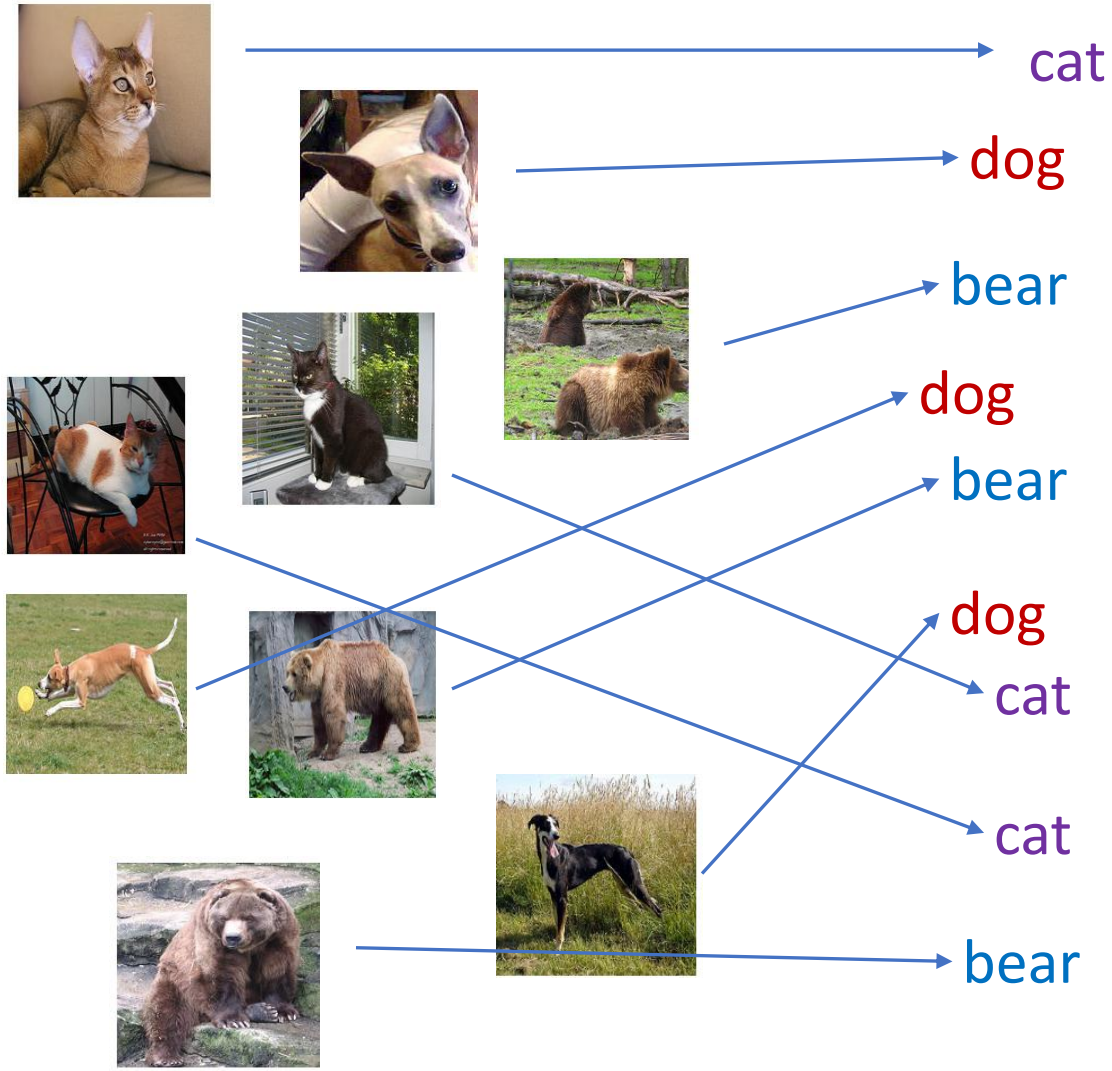
Types of ML

[Link to source]



Supervised Learning vs Unsupervised Learning

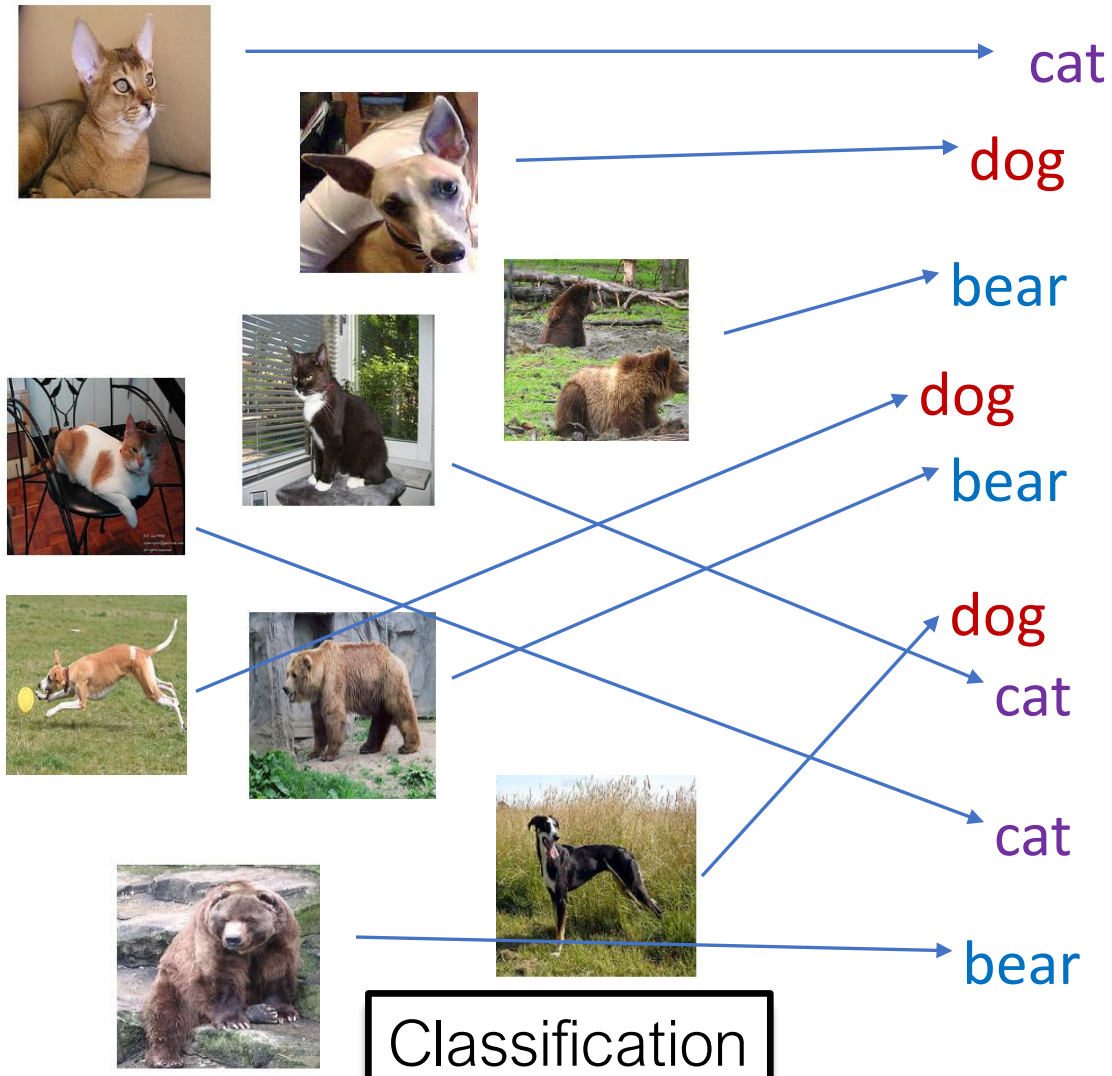
$x \rightarrow y$



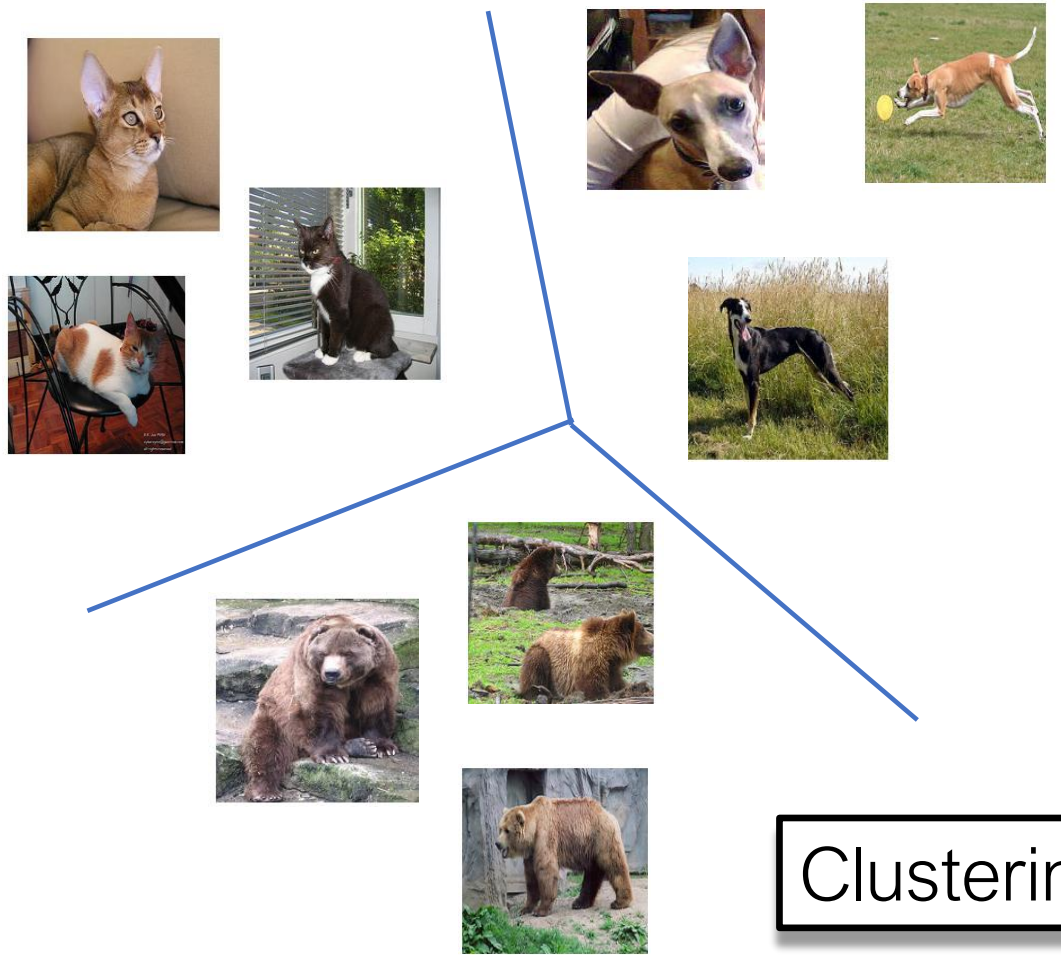
x

Supervised Learning vs Unsupervised Learning

$x \rightarrow y$

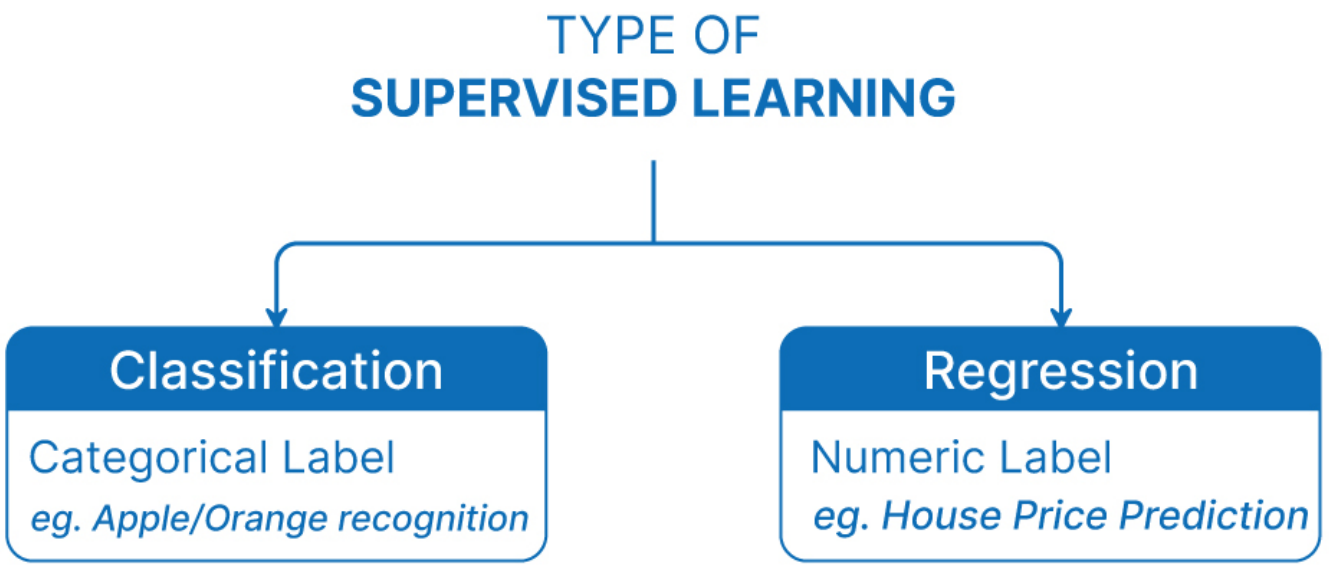


x



Supervised Learning

- Uses a set of **input values** with their **labels** to train the model
  - E.g., classify an email into spam or not spam
  - E.g., using information about a house such as the number of rooms, size, location, to predict the output variable -> the house price

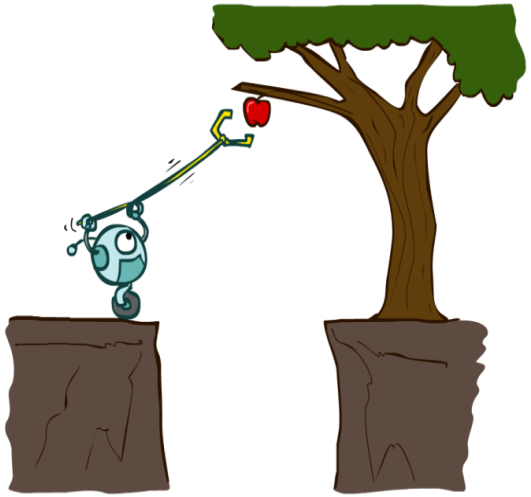


Unsupervised Learning

- Sometimes we don’t have output labels available in our dataset and we wish to group our data into clusters
  - e.g., having a movies dataset with lots of movie features such as their genre, rating, target audience, etc, and then grouping movies that are similar to each other

Reinforcement Learning: Optimization via Interaction

- Think of RL as training a dog or a gamer playing a new level **without a tutorial**
  - You don't program the rules explicitly (like 'if X, do Y')
  - Instead, you define the goal: 'Get the high score'
  - The model tries random things —running into walls (penalty), finding a coin (reward)—and over thousands of iterations, it figures out the optimal strategy to win
  - This is exactly how AlphaGo learned to beat human world champions.



Reinforcement Learning: Optimization via Interaction

- Agent & Environment: An autonomous agent observes a state and makes a decision (Action)
- Reward System: The model learns a policy based on feedback via trial and error:
  - **Reinforcement**: Positive feedback (+ Points) increases the probability of repeating the action
  - **Penalty**: Negative feedback (- Points) discourages the action.
- Iterative Improvement: The model evolves from random moves to optimal strategy through millions of simulations
- Key Example: Game AI (Chess, Go) and Robotics Control

What is Generative AI?

- **Definition:** Generative AI refers to algorithms that can create original content—including text, images, audio, video, and computer code—rather than simply analyzing existing data
  - **The Shift:** Traditional AI (like spam filters) classifies data. Generative AI learns patterns from massive datasets to produce *new* outputs that look like the training data.
- **Examples:**
  - **Text:** ChatGPT, Claude, Llama (writing essays, emails, summaries)
  - **Images:** DALL-E, Midjourney (creating art from descriptions)
  - **Code:** Writing software and fixing bugs

- Think of GenAI as a machine that doesn't just read books but can write its own stories based on what it has read

The Engine: Large Language Models (LLMs) & Transformers

- **What is an LLM?** A Large Language Model is a neural network trained on vast amounts of text to understand and generate human language
- **The Architecture:** Modern LLMs are built on the **Transformer** architecture (introduced by Google in 2017). This architecture allows the model to process entire sequences of data at once rather than word-by-word
- **The Secret Sauce: Attention.** Transformers use a mechanism called **Self-Attention**
  - This allows the model to look at every word in a sentence simultaneously and decide which words are most relevant to each other (e.g., understanding that "bank" refers to a river, not money, based on context)

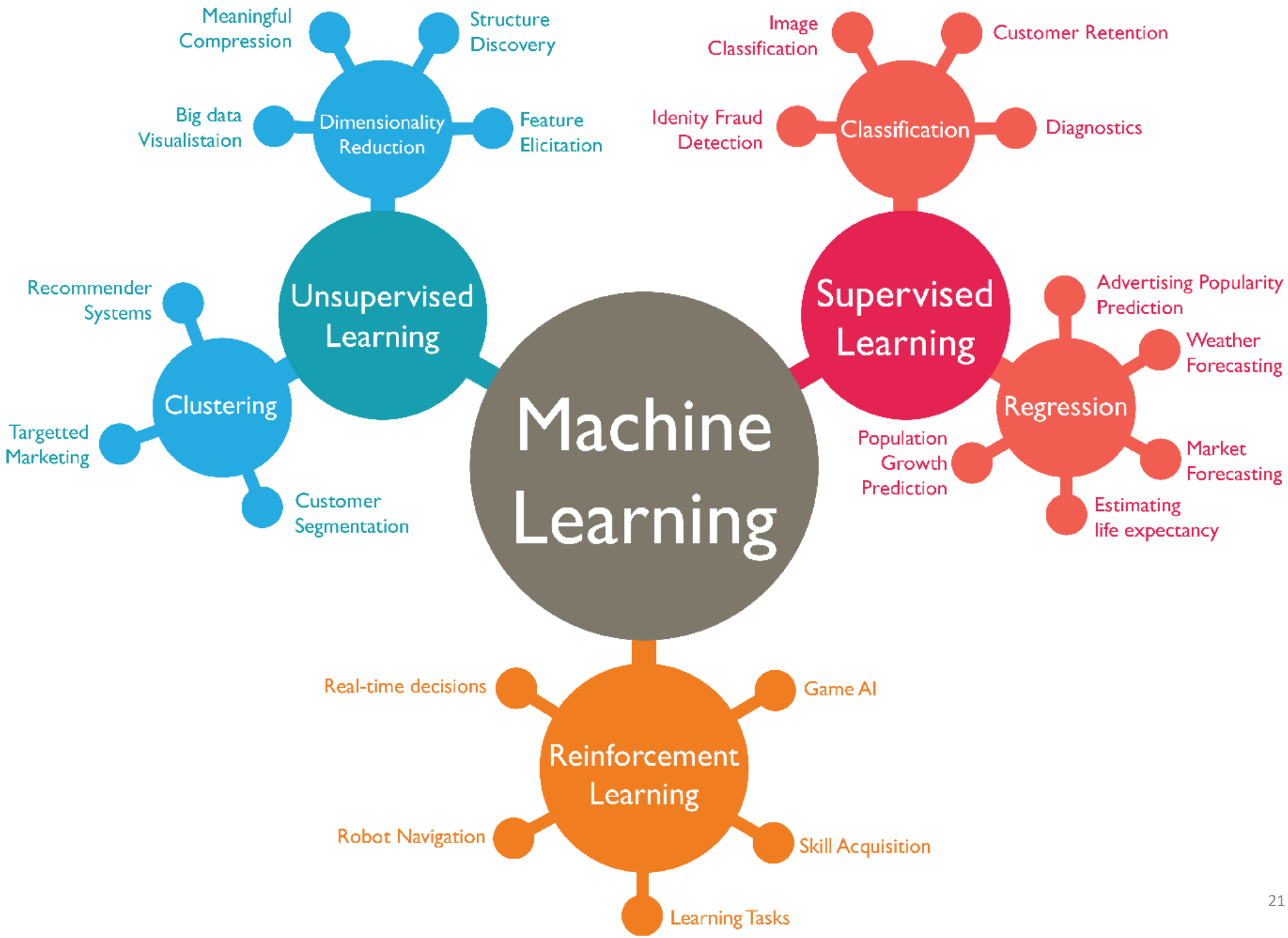
How LLMs Are Trained

- **Step 1: Pre-training (The Knowledge Base).**
  - The model is fed terabytes of text (books, websites).
  - It plays a game of "guess the next word" billions of times. This process "compresses" the internet's knowledge into the model's parameters.
- **Step 2: Fine-Tuning (The Specialization)**
  - The pre-trained model is raw. Developers feed it high-quality Q&A examples to teach it how to act like a helpful assistant.
- **Step 3: RLHF (The Safety Check).**
  - **Reinforcement Learning from Human Feedback** involves humans rating the AI's answers. The model learns to prefer answers that humans find helpful and safe.

Moving Beyond Simple Q&A

- **Prompt Engineering:** The skill of crafting clear, specific inputs (context, role, instructions) to guide the model toward the best output
- **RAG (Retrieval Augmented Generation):** Connecting the LLM to your private data (like company PDFs). The model looks up facts before answering, reducing "hallucinations" (wrong answers)
- **Agentic AI:** Instead of just talking, **Agents** can use tools. They can reason through multi-step problems, browse the web, use calculators, or execute code to complete tasks on your behalf

ML Application Areas



[Link to source]

Common Classic ML Algorithms

Link to source

Linear		Linear Regression	**The "best fit" line** through all data points. Predictions are numerical.	Easy to understand — you clearly see what the biggest drivers of the model are.	Sometimes too simple to capture complex relationships between variables. Does poorly with correlated features.
		Logistic Regression	The adaptation of **linear regression** to problems of classification (e.g., yes/no questions, groups, etc.)	Also easy to understand.	Sometimes too simple to capture complex relationships between variables. Does poorly with correlated features.
Tree-Based		Decision Tree	**A series of yes/no rules based on the features,** forming a tree, to match all possible outcomes of a decision.	Easy to understand.	Not often used on its own for prediction because it's also often too simple and not powerful enough for complex data.
		Random Forest	Takes advantage of many decision trees, with rules created from subsamples of features. Each tree is weaker than a full decision tree, but **by combining them we get better overall performance.**	A sort of "wisdom of the crowd." Tends to result in very high quality models. Fast to train.	Models can get very large. Not easy to understasnd predictions.

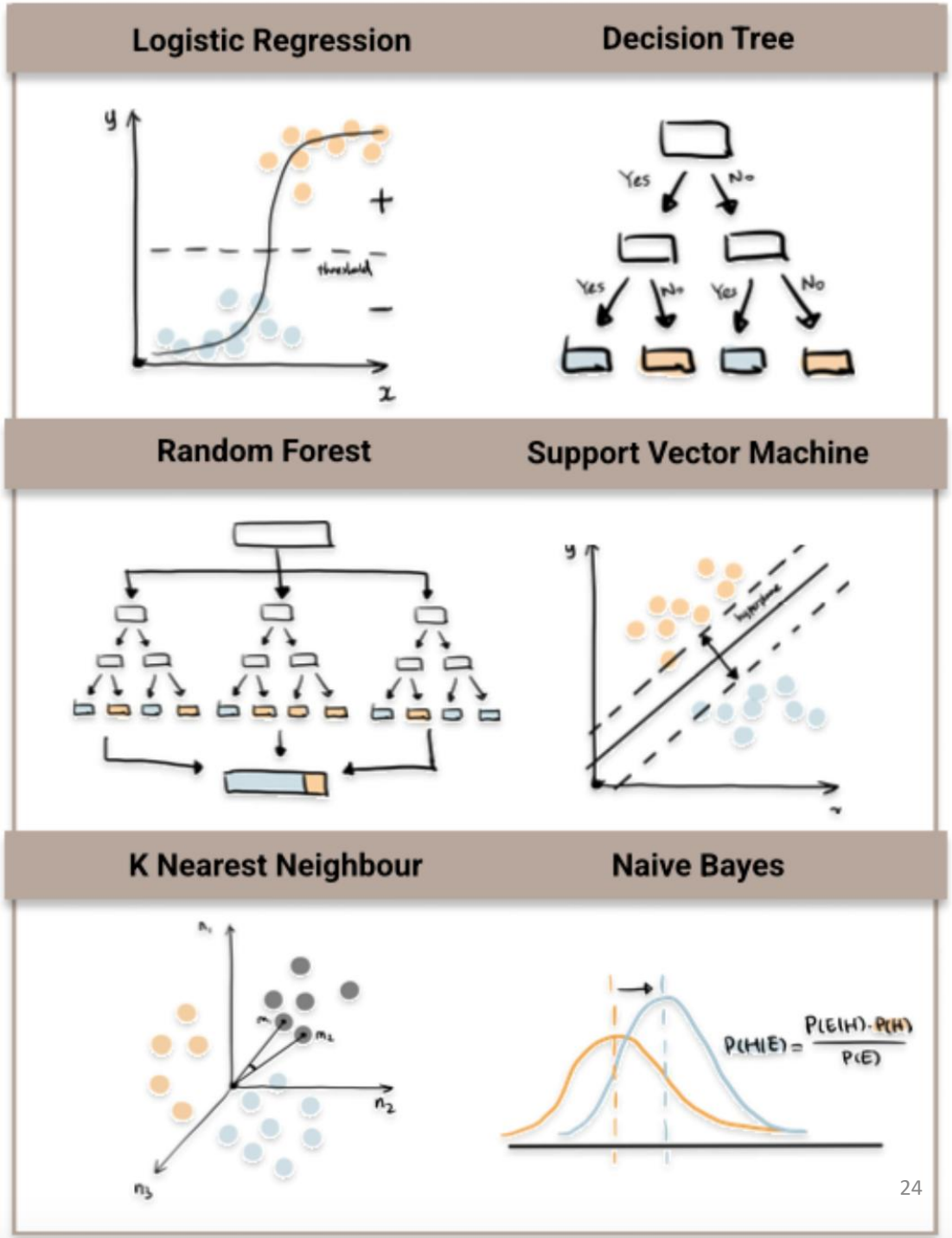
Common ANN ML Algorithms

Link to source

Neural Networks

	**CNN (Convolutional Neural Network)**	Neurons connected in a grid fashion to be spatially aware.	Best in class for image recognition use cases.	Vulnerable to noise & examples which are out of the training data context.
	**RNN (Recurrent Neural Network)**	Neurons connected sequentially for temporal awareness.	Performant for time series use cases.	Weak for long-term memory — struggle to remember patterns from much earlier time steps.
	**LSTM (Long Short-Term Memory)**	RNNs capable of both long- & short-term memory.	Can selectively remember and recall history from the recent & distant past.	Slow to compute and difficult to optimize.
	**Transformer**	Networks with the ability to focus attention on history more selectively.	Best in class for natural language understanding, processing, and generation use cases.	Very resource intensive, even for inference. Very difficult to interpret.

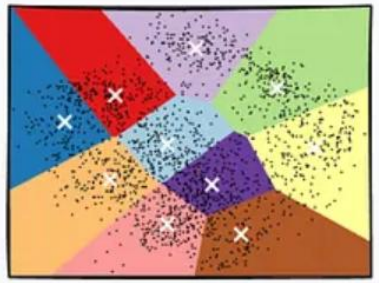
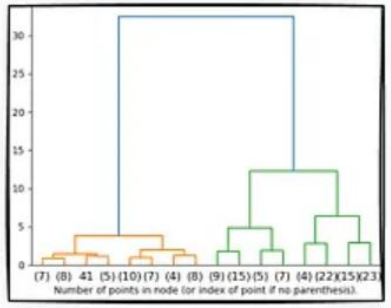
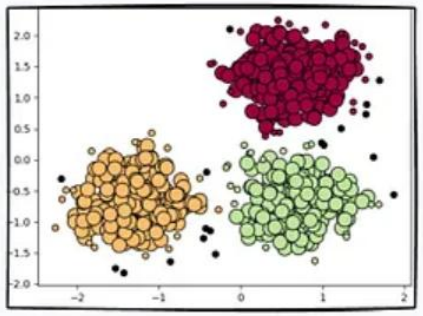
Common Classification Algorithms



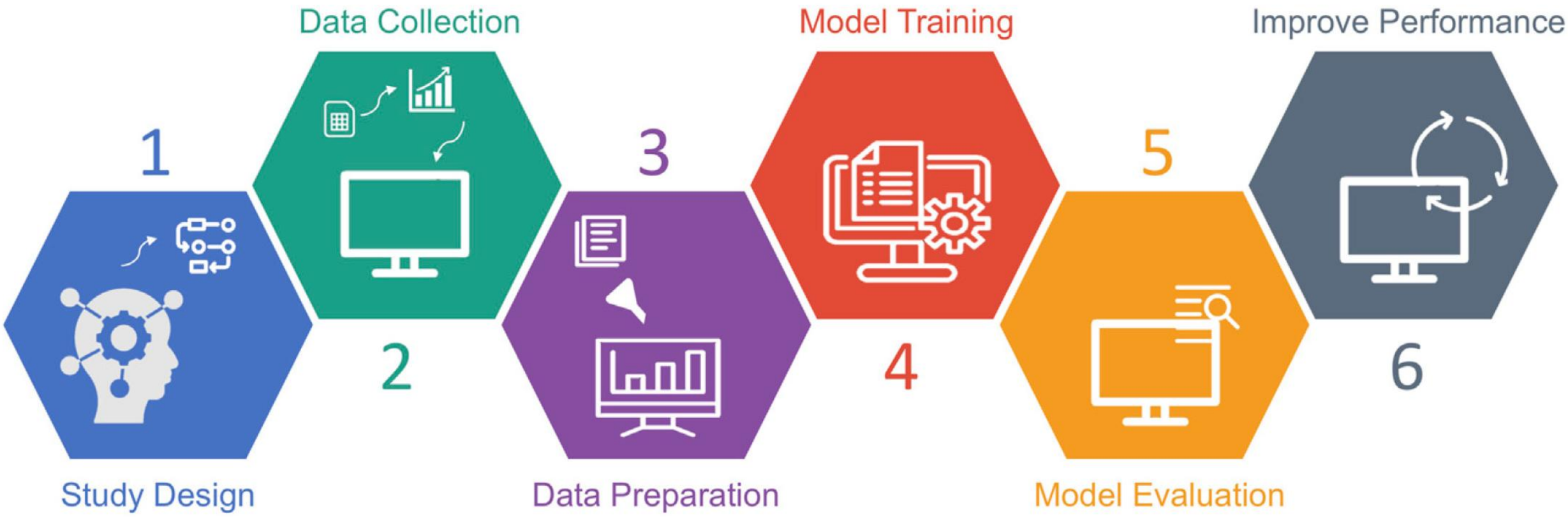
[Link to source]

Common Clustering Algorithms

Link to source

Clustering Algorithm Type		Clustering Methodology	Algorithm(s)
	Centroid-based	Cluster points based on proximity to centroid	KMeans KMeans++ KMedoids
	Connectivity-based	Cluster points based on proximity between clusters	Hierarchical Clustering (Agglomerative and Divisive)
	Density-based	Cluster points based on their density instead of proximity	DBSCAN OPTICS HDBSCAN

ML Workflow



Link to source

Applications Areas

- **Natural Language Processing (NLP):** Question answering, speech recognition, Document summarization and classification, Text-to-speech and speech synthesis
- **Computer Vision:** Face recognition and image captioning, Traffic sign recognition, Pedestrian and vehicle detection for autonomous driving
- **Image Generation & Enhancement:** Image colorization and super-resolution, Denoising and restoration
- **Medicine:** Anomaly detection in CT, MRI, and X-ray images
- **Recommendation Systems:** Web search and ranking, Product recommendations
- **Robotics:** Perception and manipulation of objects
- Many emerging real-world AI systems

Resources

- Cheat Sheets of Machine Learning

https://medium.com/machine-learning-in-practice/cheat-sheet-of-machine-learning-and-python-and-math-cheat-sheets-a4afe4e791b6