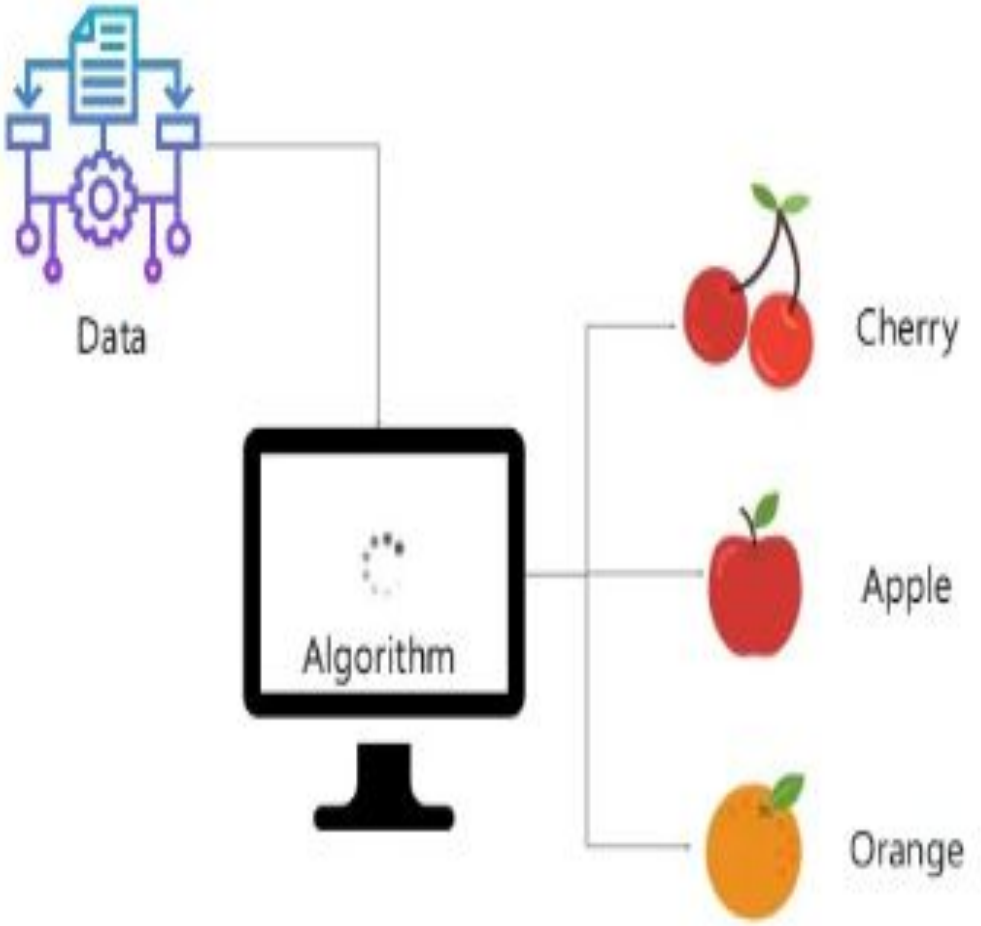
Introduction to Machine Learning

Module I: Introduction to Machine Learning

Introduction to Machine Learning: Definition and significance of ML. Understanding Machine Learning, Machine learning paradigms-supervised, semi-supervised, unsupervised, reinforcement learning. Programming Languages for AI and ML: Python Libraries: Pandas, NumPy, Matplotlib, seaborn, Scikit-learn.

What Is Machine Learning?

Machine learning is a subset of artificial intelligence (AI) which provides machines the ability to learn automatically & improve from experience without being explicitly programmed.



Definition and significance of ML

- **Machine learning** is a branch of Artificial Intelligence that focuses on developing models and algorithms that let computers learn from data without being explicitly programmed for every task.
- In simple words, ML teaches the systems to think and understand like humans by learning from the data

Machine Learning

- The term machine learning was first introduced by Arthur Samuel in 1959. We can define it in a summarized way as:
- Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed.

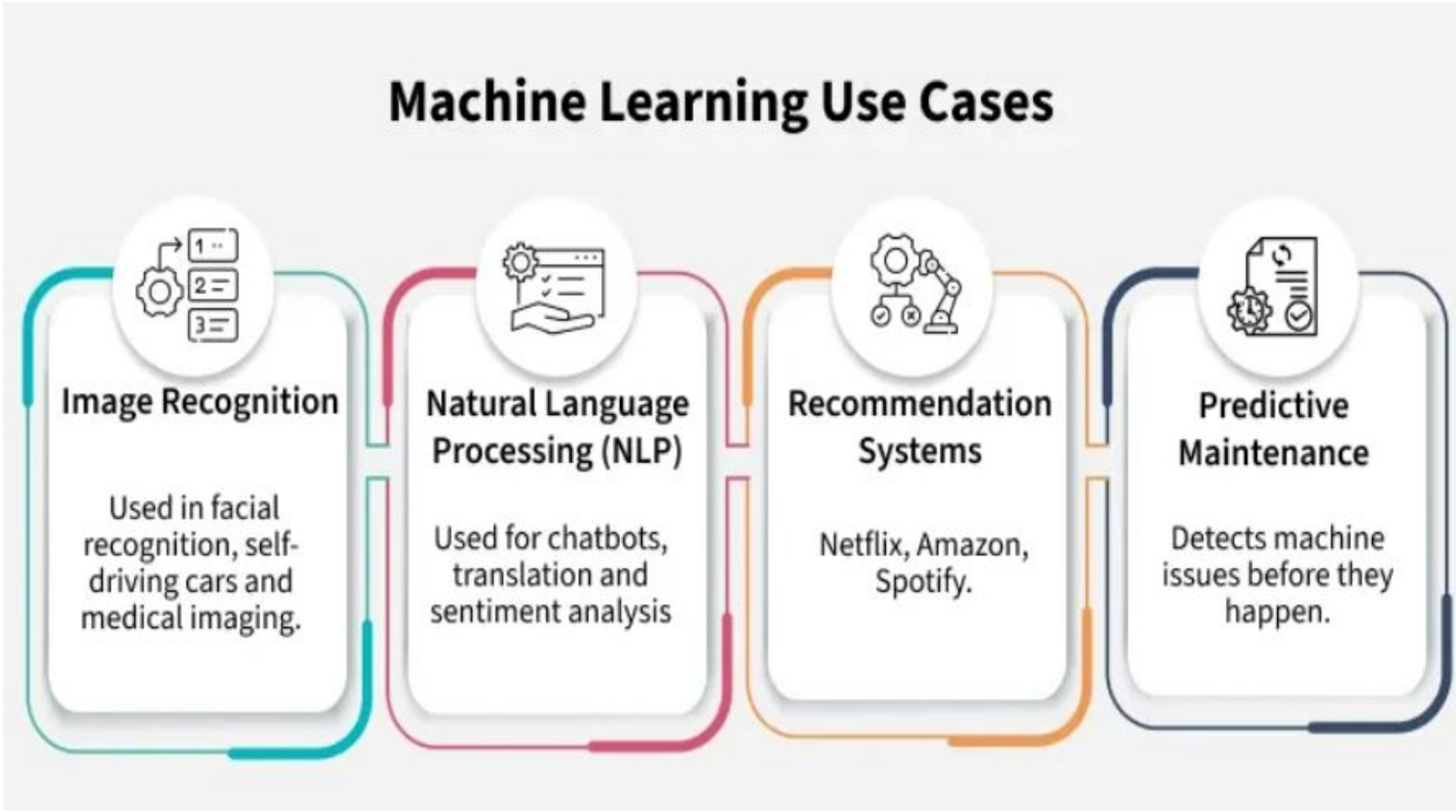
Formal Definition (Tom Mitchell, 1997):

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

- **Task (T)**: What the model is supposed to do (e.g., classify emails as spam or not).
- **Performance (P)**: How well it does the task (e.g., accuracy).
- **Experience (E)**: The data used for learning (e.g., historical emails).

Using data for answering questions

Training Predicting


Machine Learning Use Cases
Image Recognition
Used in facial recognition, self-driving cars and medical imaging.
Natural Language Processing (NLP)
Used for chatbots, translation and sentiment analysis
Recommendation Systems
Netflix, Amazon, Spotify.
Predictive Maintenance
Detects machine issues before they happen.

Understanding machine learning

Step 1: Data collection

Step 2: Data preprocessing

Step 3: Choosing the right model

Step 4: Training the model

Step 5: Evaluating the model

Step 6: Hyperparameter tuning and optimization

Step 7: Predictions and deployment

Step 1: Data collection

- Data is the lifeblood of machine learning - the quality and quantity of your data can directly impact your model's performance.
- Data can be collected from various sources such as databases, text files, images, audio files, or even scraped from the web.
- Once collected, the data needs to be prepared for machine learning.
- This process involves organizing the data in a suitable format, such as a CSV file or a database, and ensuring that the data is relevant to the problem you're trying to solve.

Step 2: Data preprocessing

- Data preprocessing is a crucial step in the machine learning process.
- It involves cleaning the data (removing duplicates, correcting errors), handling missing data (either by removing it or filling it in), and normalizing the data (scaling the data to a standard format).
- Preprocessing improves the quality of your data and ensures that your machine learning model can interpret it correctly.
- This step can significantly improve the accuracy of your model.

Step 3: Choosing the right model

- Once the data is prepared, the next step is to choose a machine learning model.
- There are many types of models to choose from, including linear regression, decision trees, and neural networks.
- The choice of model depends on the nature of your data and the problem you're trying to solve.

Step 4: Training the model

- After choosing a model, the next step is to train it using the prepared data.
- Training involves feeding the data into the model and allowing it to adjust its internal parameters to better predict the output.
- During training, it's important to avoid **overfitting** (where the model performs well on the training data but poorly on new data) and **underfitting** (where the model performs poorly on both the training data and new data).

Step 5: Evaluating the model

- Once a model is trained, evaluating its performance on unseen data is essential before deployment.
- Common metrics for evaluating a model's performance include accuracy (for classification problems), precision and recall (for binary classification problems), and mean squared error (for regression problems).

Step 6: Hyperparameter tuning and optimization

- When training a machine learning model, there are certain settings called **hyperparameters** (like learning rate, number of layers, or number of neighbors in KNN) that we must choose before training. These affect how well the model learns.

Why Tune Hyperparameters?

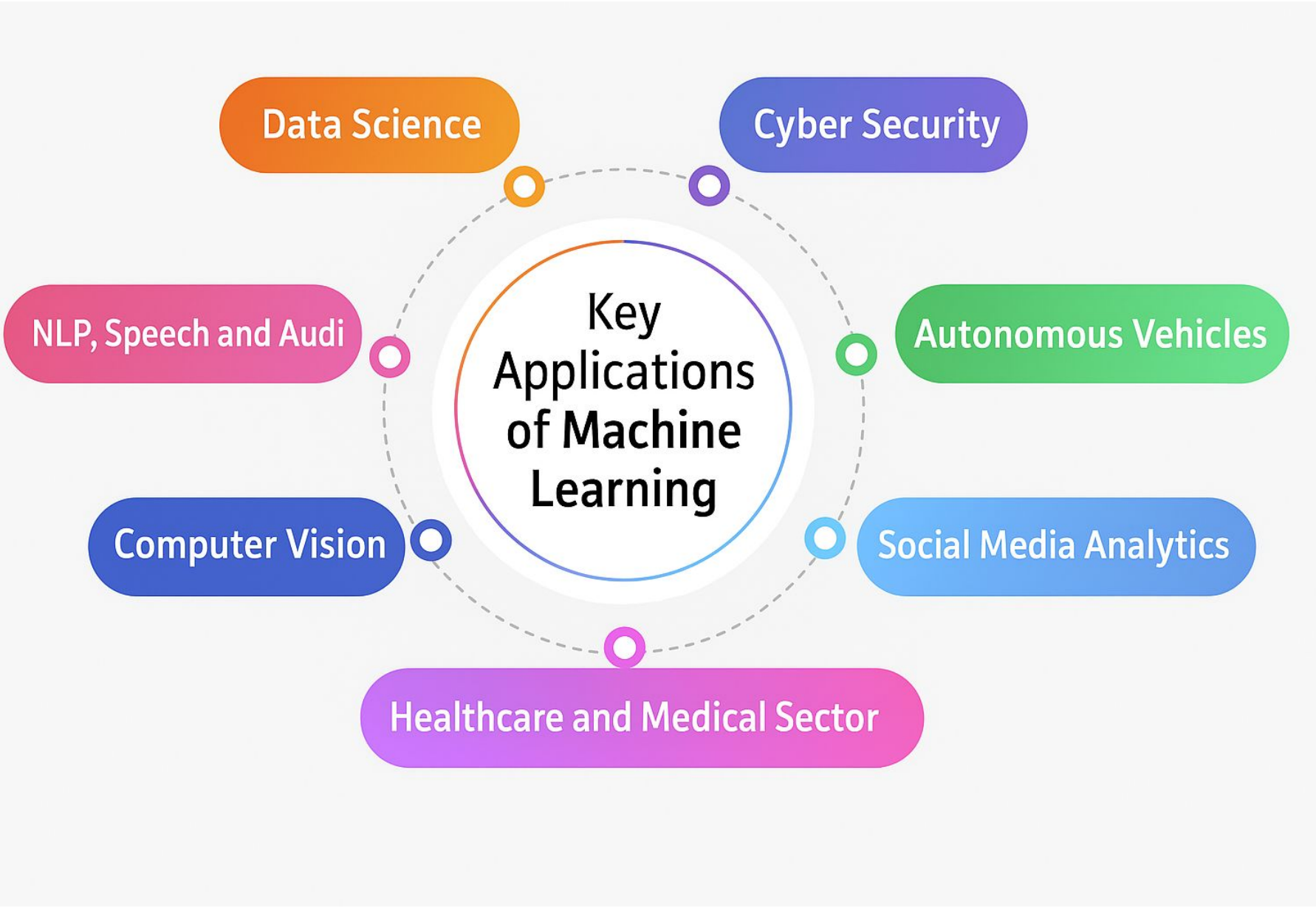
To make sure the model works well, we need to find the **best combination** of these settings.

Simple Techniques:

- **Grid Search**: Try out all possible combinations of hyperparameter values and see which one works best.
- **Cross-Validation**: Split your data into parts and train the model on different parts to make sure it performs well on any data, not just the training set.

Step 7: Predictions and deployment

- After training a machine learning model on data, you can use it to **make predictions on new data** it hasn't seen before.
- Deployment means **making your model available for real-world use —** so other apps, websites, or people can send input to it and get predictions.
- Continuously monitor the model to check its performance and accuracy over time.


Key Applications of Machine Learning
Data Science
Cyber Security
NLP, Speech and Audi
Autonomous Vehicles
Computer Vision
Social Media Analytics
Healthcare and Medical Sector

Machine learning paradigms

- Machine Learning is mainly divided into three core types: Supervised, Unsupervised and Reinforcement Learning along with two additional types, Semi-Supervised and Self-Supervised Learning.

Types Of Machine Learning

Supervised Learning

Unsupervised Learning

Reinforcement Learning

Supervised learning is a method in which we teach the machine using labelled data

In unsupervised learning the machine is trained on unlabelled data without any guidance

In Reinforcement learning an agent interacts with its environment by producing actions & discovers errors or rewards

Supervised Learning

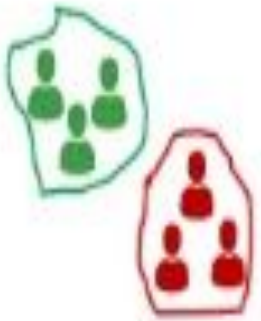
Regression

Classification

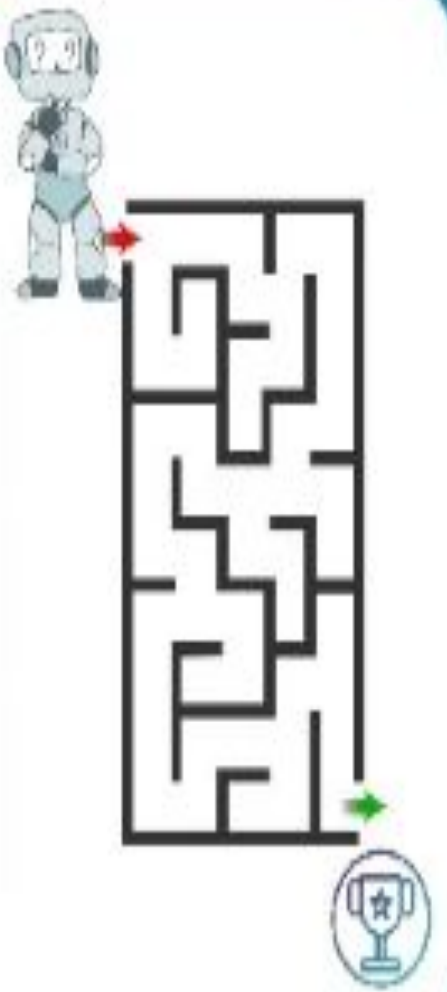
Unsupervised Learning

Association

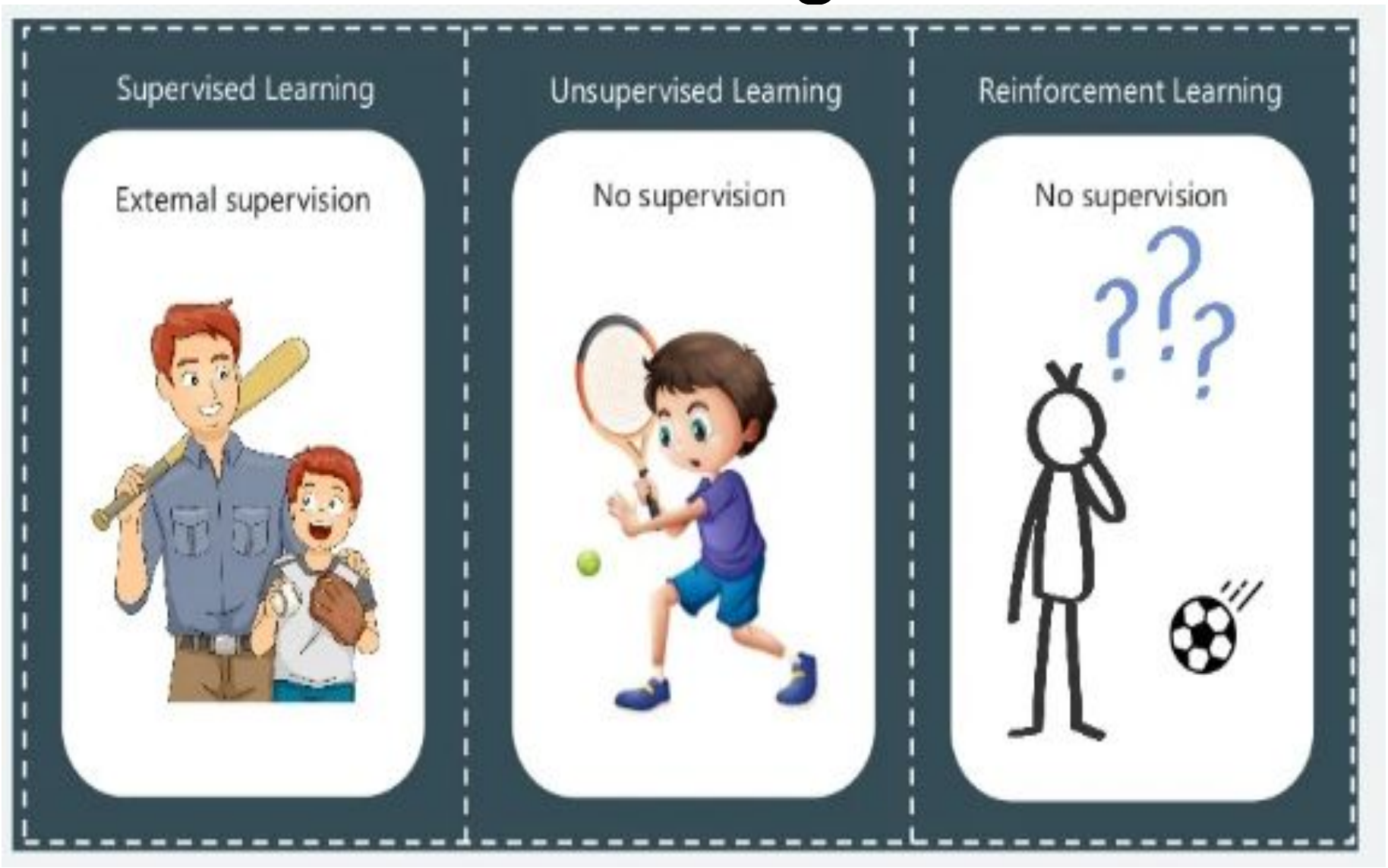
Clustering

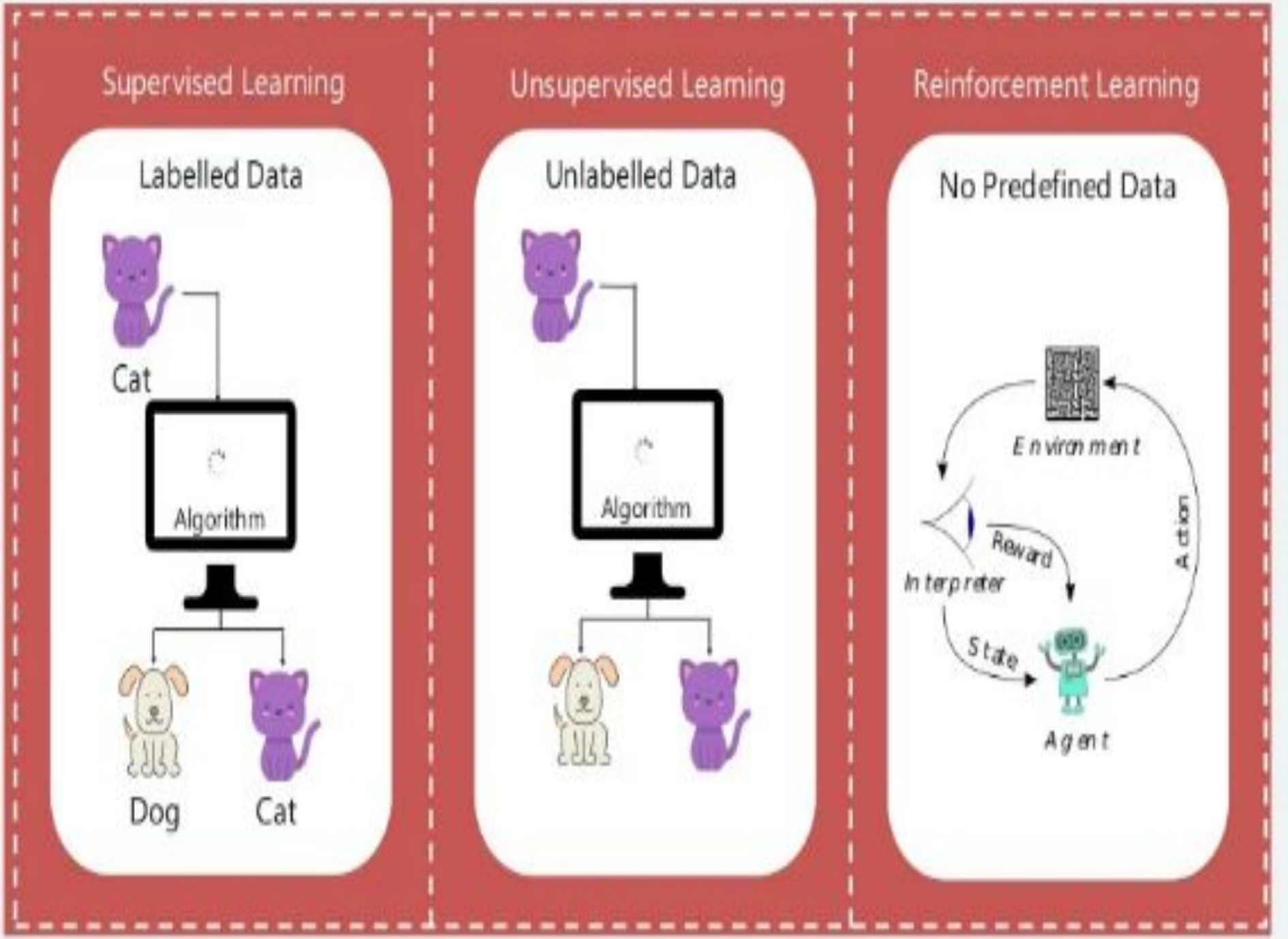


Reinforcement Learning



Training


Supervised Learning
External supervision
Unsupervised Learning
No supervision
Reinforcement Learning
No supervision


Supervised Learning
Labelled Data
Cat
Algorithm
Dog
Cat
Unsupervised Learning
Unlabelled Data
Algorithm
Reinforcement Learning
No Predefined Data
Environment
Reward
Interpreter
Action
State
Agent

Supervised Learning

Map labelled input to known output

Labelled Input

Training

Algorithm

Known Output

Unsupervised Learning

Understand patterns and discover output

Unlabelled Input

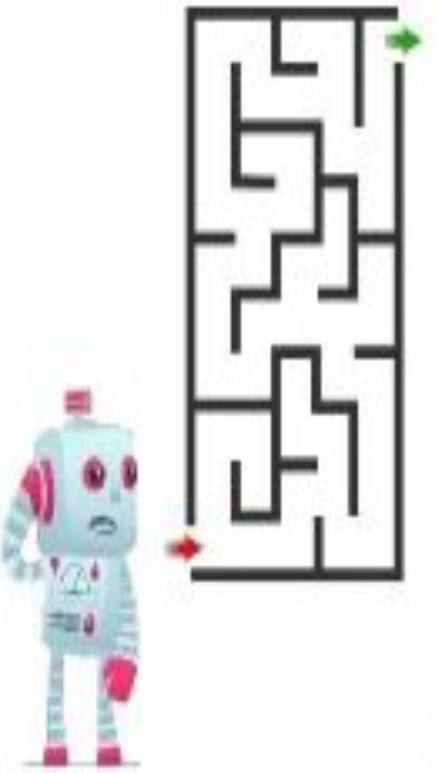
Explore patterns & trends

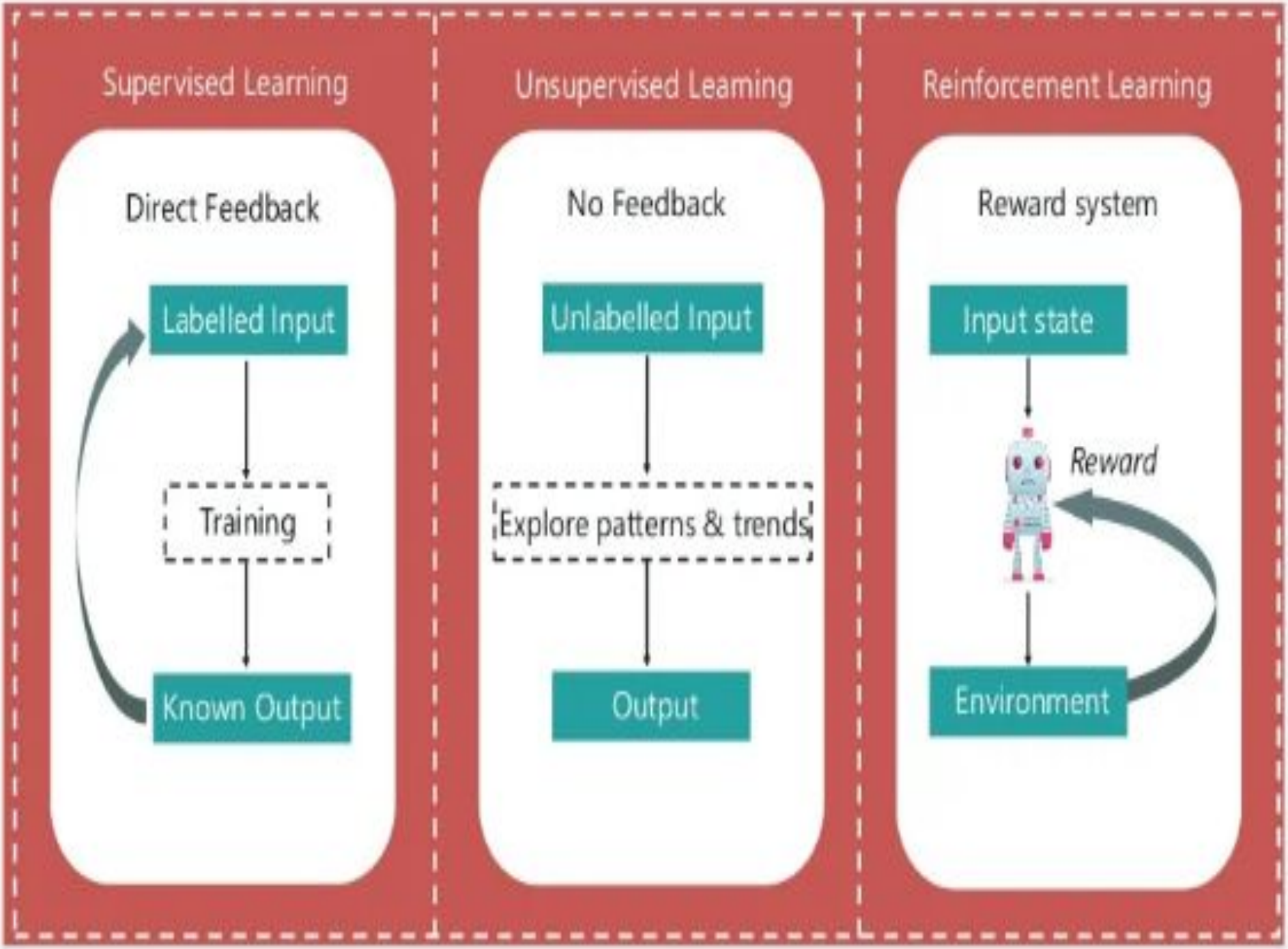
Algorithm

Output

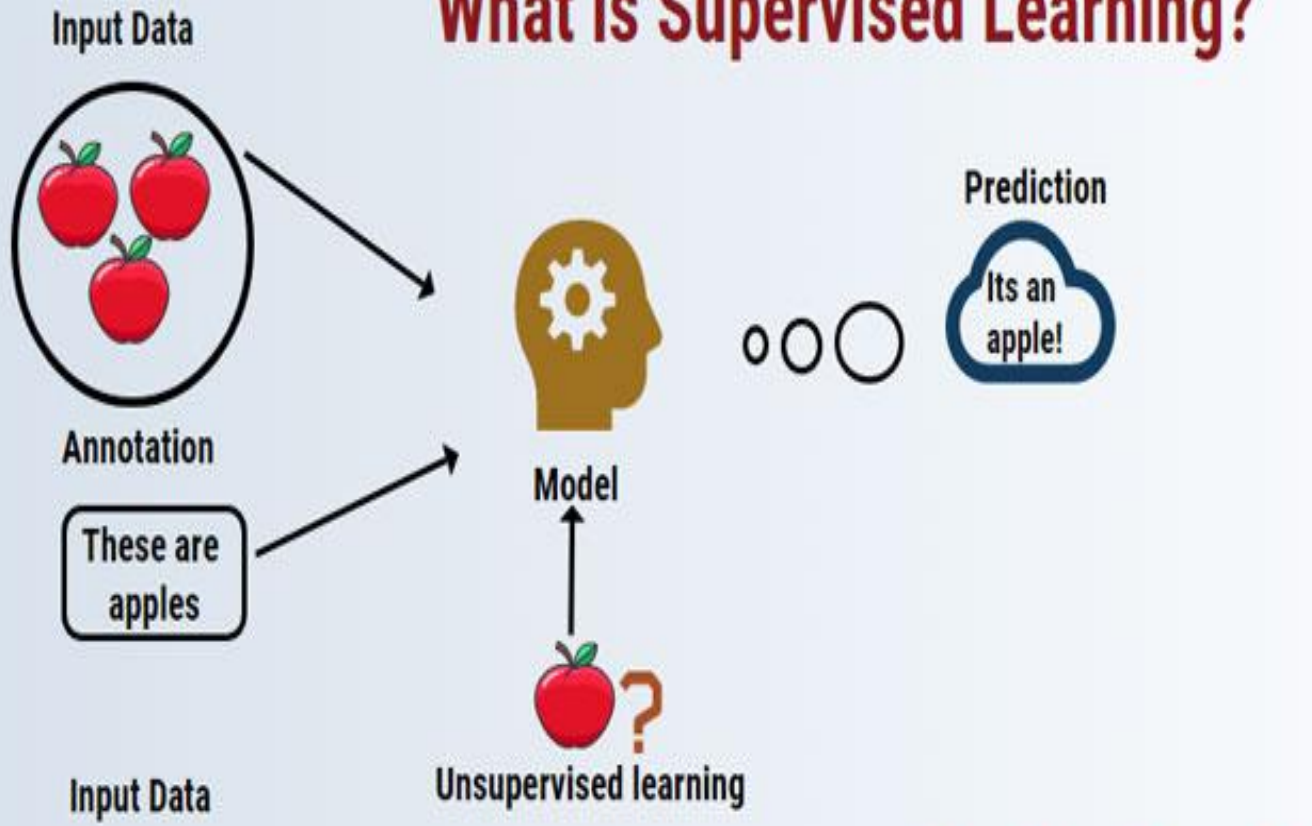
Reinforcement Learning

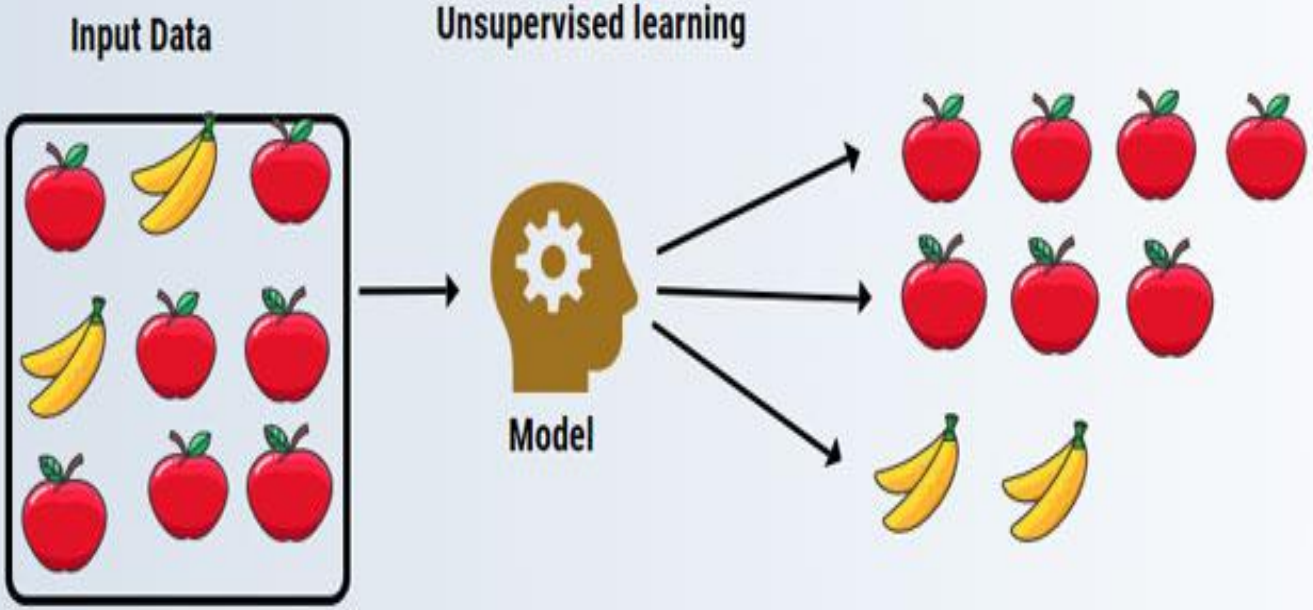
Follow Trail and Error method




Supervised Learning
Direct Feedback
Labelled Input
Training
Known Output
Unsupervised Learning
No Feedback
Unlabelled Input
Explore patterns & trends
Output
Reinforcement Learning
Reward system
Input state
Reward
Environment

What is Supervised Learning?





- Supervised learning is like learning with a teacher. The model is trained on a labeled dataset, meaning each input has a corresponding output. The key characteristics of supervised learning are:
- **Labeled Data:** Training data has predefined labels.
- **Example Problems:** Used for spam detection and for predicting house prices.
- **Algorithms:** Linear Regression, Logistic Regression, SVM, Decision Trees, Neural Networks.

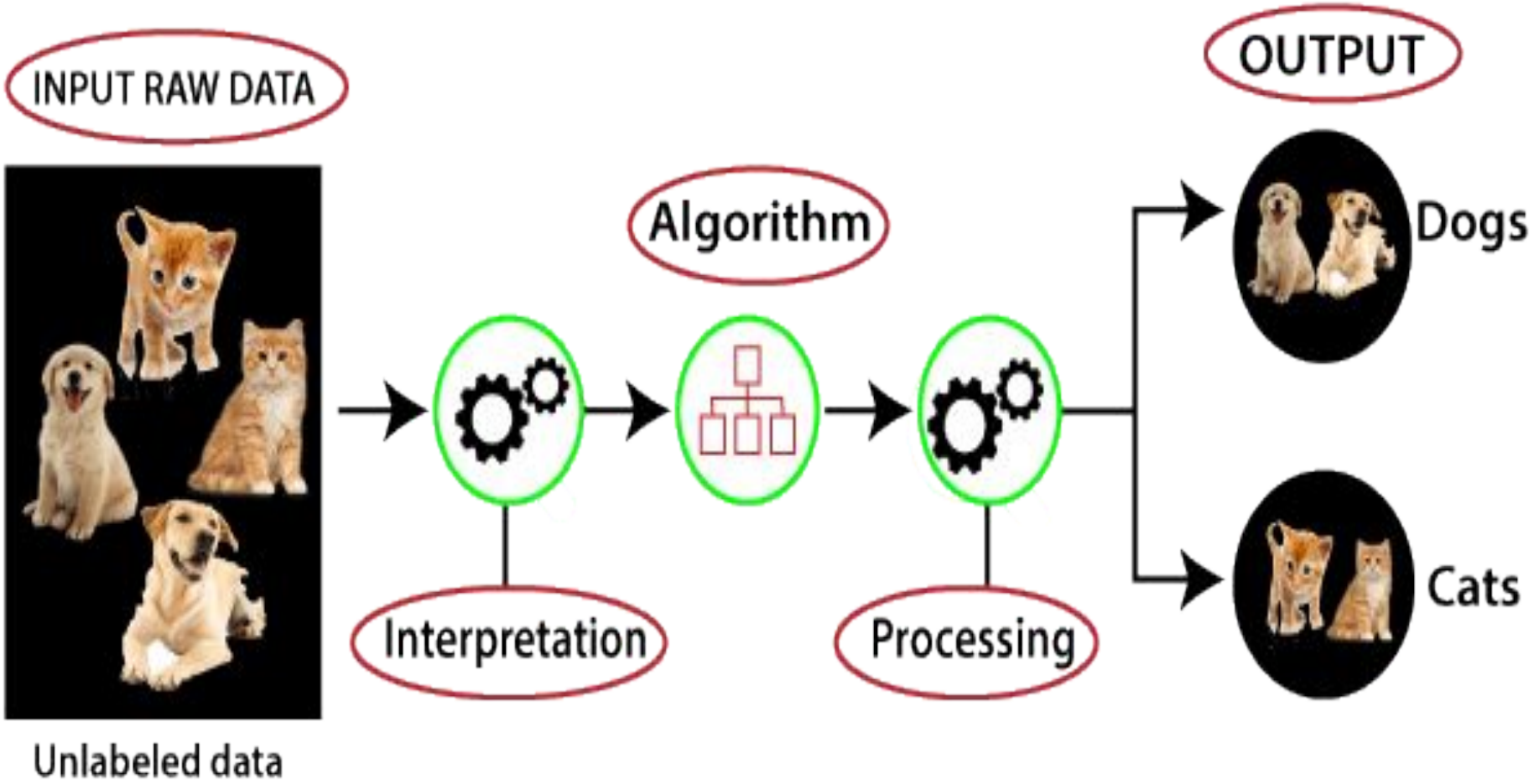
Unsupervised Learning : Finds patterns or groups in unlabeled data. It works with data that has no predefined labels. The model identifies patterns, clusters or associations independently.

The key characteristics of unsupervised learning are:

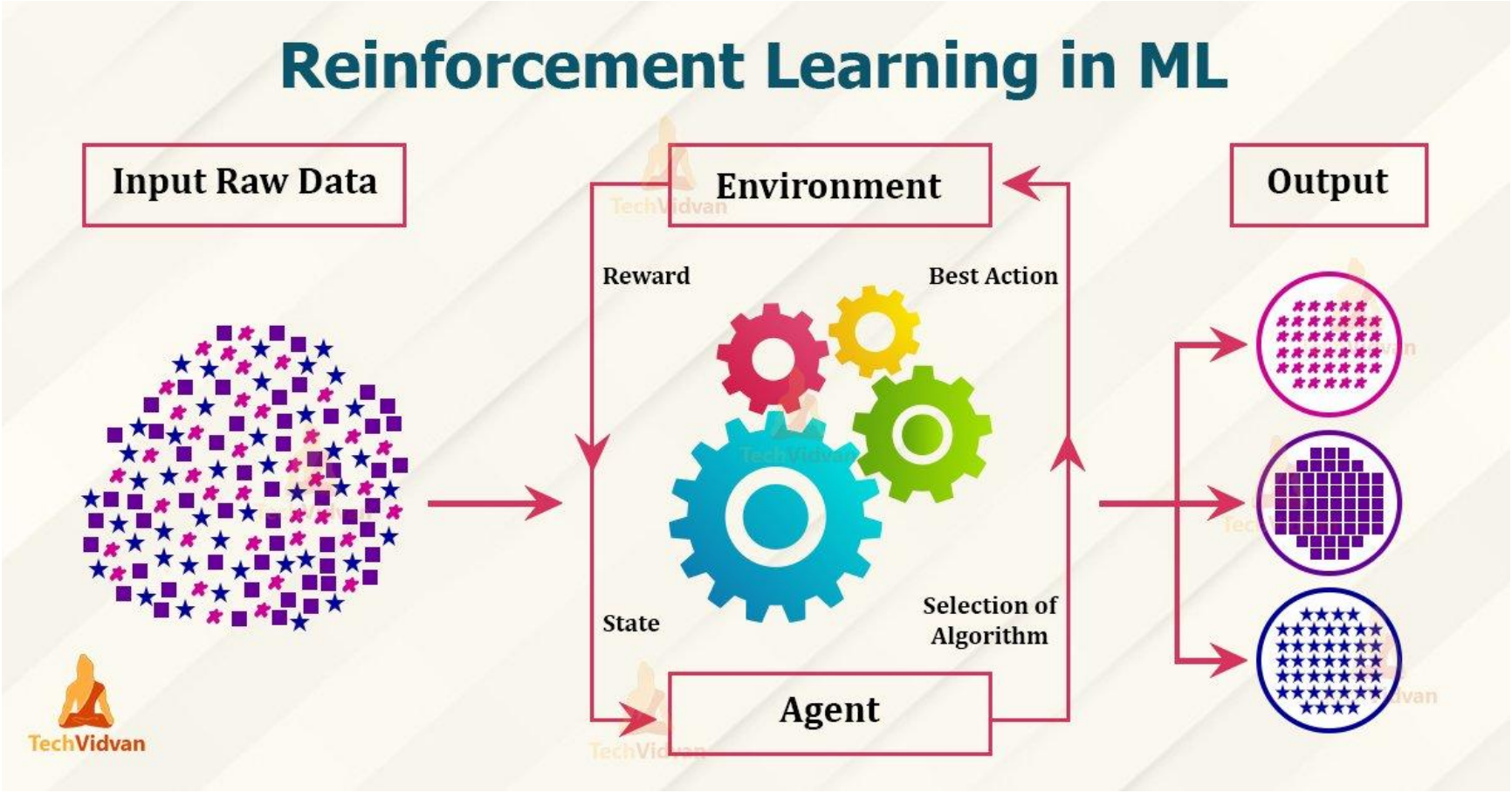
Unlabeled Data: No predefined outputs.

Types of Problems: Used for **tasks like** customer segmentation and for market basket analysis.

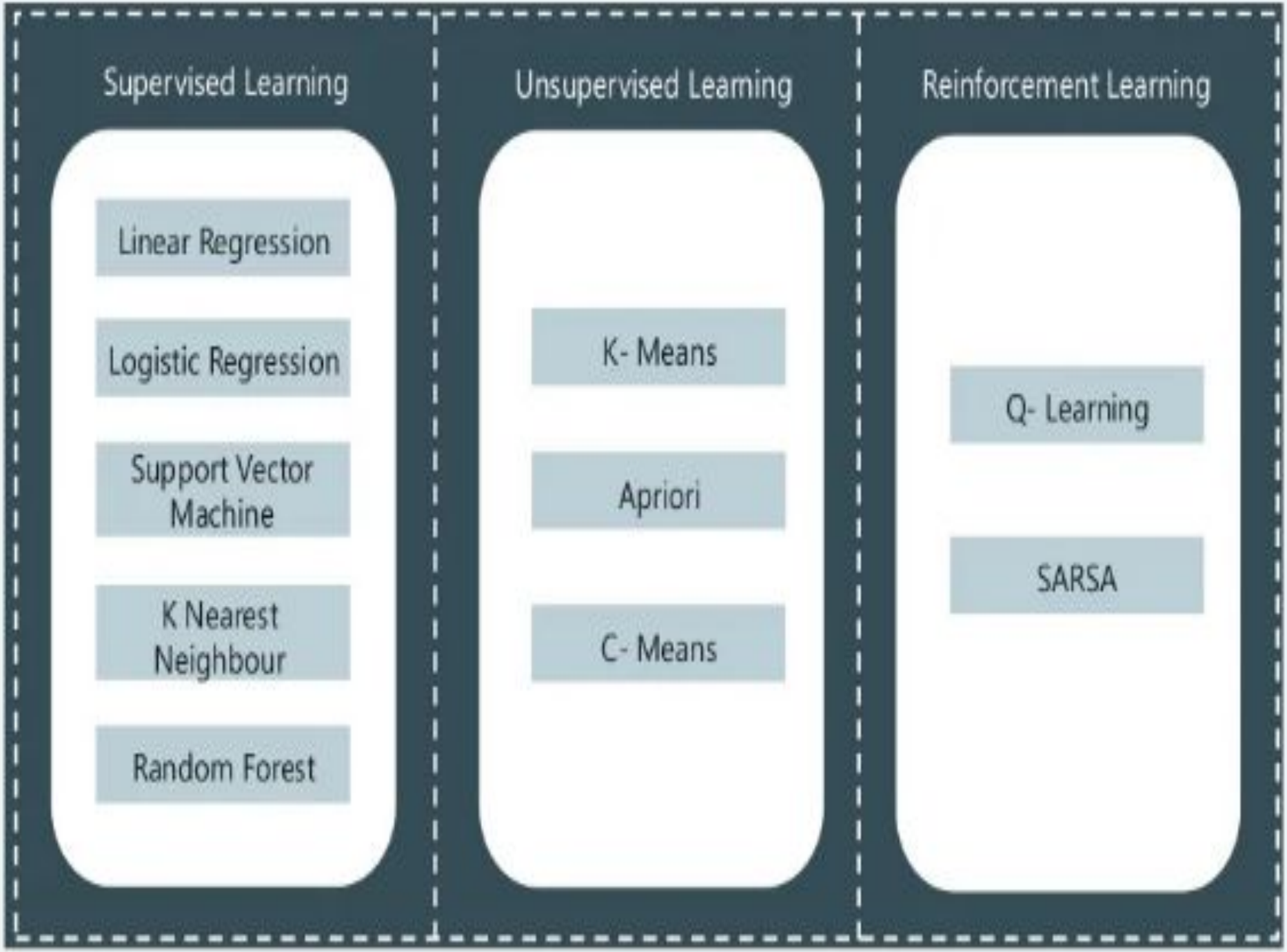
Algorithms: K-Means, Hierarchical Clustering, PCA, Autoencoders.


INPUT RAW DATA
Unlabeled data
Algorithm
Interpretation
Processing
OUTPUT
Dogs
Cats

Reinforcement Learning : Learns through trial and error to maximize rewards, ideal for decision-making tasks.



- Reinforcement Learning revolves around the idea that an agent (the learner or decision-maker) interacts with an environment to achieve a goal. The agent performs actions and receives feedback to optimize its decision-making over time.
- **Agent**: The decision-maker that performs actions.
- **Environment**: The world or system in which the agent operates.
- **State**: The situation or condition the agent is currently in.
- **Action**: The possible moves or decisions the agent can make.
- **Reward**: The feedback or result from the environment based on the agent's action.


Supervised Learning
Linear Regression
Logistic Regression
Support Vector Machine
K Nearest Neighbour
Random Forest
Unsupervised Learning
K- Means
Apriori
C- Means
Reinforcement Learning
Q- Learning
SARSA

Supervised Learning

Forecast outcomes

Unsupervised Learning

Discover underlying patterns

Reinforcement Learning

Learn series of action

Supervised Learning

Risk Evaluation

Forecast Sales

Unsupervised Learning

Recommendation Systems

Anomaly Detection

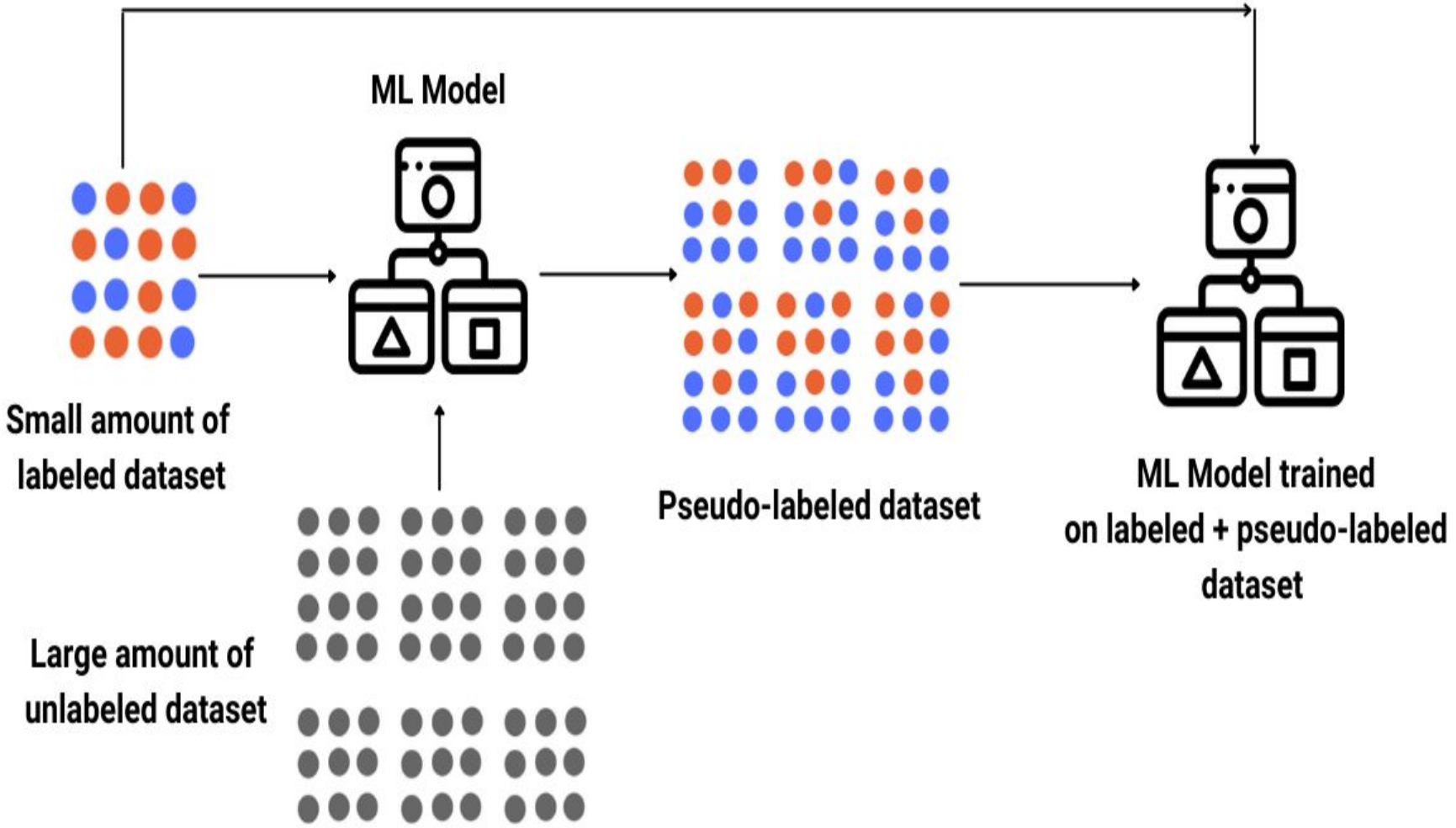
Reinforcement Learning

Self driving cars

Gaming

- **Self Supervised Learning**: Self-supervised learning is often considered a subset of unsupervised learning. It generates its own labels from the data, without any manual labeling.
- **Semi Supervised Learning :** This approach combines a small amount of labeled data with a large amount of unlabeled data. It's useful when labeling data is expensive or time-consuming.

Semi-supervised learning use-case



Programming Languages for AI and ML

- **Python** is the most widely used programming language for Artificial Intelligence (AI) and Machine Learning (ML) because of its simplicity, large community support, and powerful libraries.

Popular Python Libraries for AI/ML:

- Pandas
- NumPy
- Matplotlib
- seaborn
- Scikit-learn

Pandas

- a Python library.
- used to analyze data.
- Python library used for working with data sets.
- It has functions for analyzing, cleaning, exploring, and manipulating data.
- Allows us to analyze big data and make conclusions based on statistical theories.

Pandas gives you answers about the data. Like:

- Is there a correlation between two or more columns?
- What is average value?
- Max value?
- Min value?

Pandas are also able to delete rows that are not relevant, or contains wrong values, like empty or NULL values. This is called *cleaning* the data.

- W3Schools Pandas Tutorial

NumPy

- **NumPy** is a Python library.
- **NumPy** is used for working with arrays.
- **NumPy** is short for "Numerical Python".
- NumPy is a Python library used for working with arrays.
- It also has functions for working in domain of linear algebra, fourier transform, and matrices.
- NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.

- In Python we have lists that serve the purpose of arrays, but they are slow to process.
- NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.
- NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently.
- This behavior is called locality of reference in computer science.
- This is the main reason why NumPy is faster than lists.
- NumPy Getting Started

Matplotlib

- Matplotlib is a low level graph plotting library in python that serves as a visualization utility.
- Matplotlib was created by John D. Hunter.
- Matplotlib is open source and we can use it freely.
- Matplotlib is mostly written in python, a few segments are written in C, Objective-C and Javascript for Platform compatibility.
- Matplotlib Pyplot

Seaborn

- Seaborn is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions.
- Seaborn

Scikit-learn.

- Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python.
- It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python
- Scikit-learn Introduction

- Python Full Course | Python for Beginners (Full Course) | Best Python Course with Examples | Edureka - YouTube

Advantages of NumPy over Python Lists

1. Faster Computation

NumPy is much faster for numerical operations.

(no need for loops).

2. Less Memory Usage

NumPy arrays consume less memory than Python lists.

Stores data in a compact, fixed-type format.

3. Supports Vectorized Operations

Can perform operations on entire arrays (e.g., a + b, a * 2) directly.

No need to write for loops for element-wise processing.

4. Built-in Mathematical Functions

Provides a wide range of functions: mean(), sum(), std(), dot(), sqrt(), etc.

Python lists require manual implementation or looping.

5. Multidimensional Array Support

Easily handles 1D, 2D (matrix), 3D, or more.

Lists require nested structures, which are harder to manage.

6. Broadcasting

Allows operations between arrays of different shapes (e.g., adding scalar to array).

Python lists don't support this behavior naturally.

7. Convenient for Scientific and Statistical Computations

Ideal for data science, machine learning, image processing, etc.

Used as the base for libraries like Pandas, TensorFlow, and SciPy.

8. Data Type Control

You can specify and enforce data types (int32, float64, etc.).

Lists can store mixed types, which slows down computation.

9. Slicing and Indexing Features

Advanced indexing, slicing, filtering using conditions (arr[arr > 5]).

More powerful than list slicing.

10. Interoperability

Works efficiently with other data science libraries and C/Fortran code.