Course Code: CSE3002
Course Name: Artificial Intelligence

Faculty Name:
Dr. D Ramkumar
AP/SCOPE
VIT-AP

Module-5

Module 5:	Planning And Learning	8 Hours

Basic plan generation systems - Strips -Advanced plan generation systems – K strips -Strategic explanations – Explanation bases Learning- Machine learning, adaptive Learning. Reinforcement learning- Genetic algorithms

Planning

- Planning refers to the process of computing several steps of a problem solving before executing any of them.
- Planning is useful as a problem solving technique for non decomposable problem.

Components of Planning System:

- In any general problem solving systems, elementary techniques to perform following functions are required
  - Choose the best rule (based on heuristics) to be applied
  - Apply the chosen rule to get new problem state
  - Detect when a solution has been found
  - Detect dead ends so that new directions are explored.

Planning

VIT-AP UNIVERSITY

To choose the rules ,

- first isolate a set of differences between the desired goal state and current state, identify those rules that are relevant to reducing these difference,
- if more rules are found then apply heuristic information to choose out of them.

To apply rules,

- In simple problem solving system,
  - applying rules was easy as each rule specifies the problem state that would result from its application.
  - In complex problem we deal with rules that specify only a small part of the complete problem state.

What is planning in AI?

VIT-AP UNIVERSITY

- Planning in artificial intelligence is about decision-making actions performed by robots or computer programs to achieve a specific goal.

- Execution of the plan is about choosing a sequence of tasks with a high probability of accomplishing a specific task.

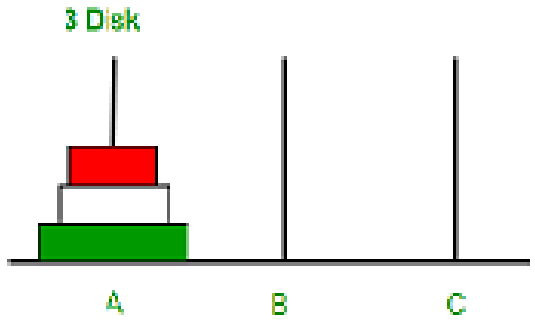
Let us consider the famous problem name as Tower of Hanoi,which helps to understand the importance of planning in artificial intelligent system.

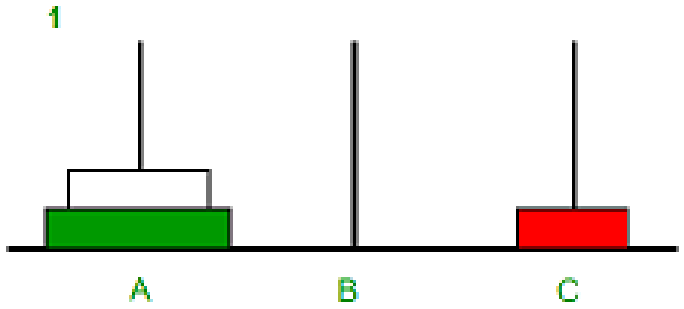
Tower of Hanoi is a mathematical puzzle where we have three rods (A, B, and C) and N disks. Initially, all the disks are stacked in decreasing value of diameter i.e., the smallest disk is placed on the top and they are on rod A. The objective of the puzzle is to move the entire stack to another rod (here considered C), obeying the following simple rules:

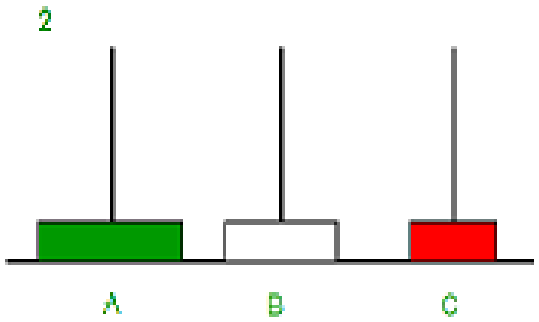
- Only one disk can be moved at a time.
- Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.
- No disk may be placed on top of a smaller disk.

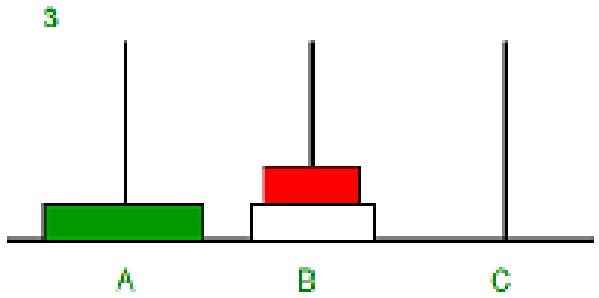
Components of Planning System:

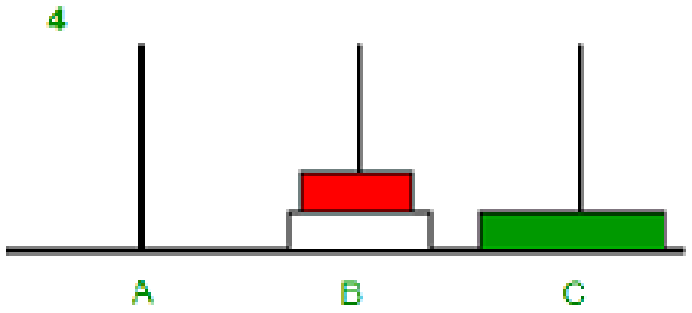
- Choose the best rule (based on heuristics) to be applied
- Apply the chosen rule to get new problem state
- Detect when a solution has been found
- Detect dead ends so that new directions are explored.

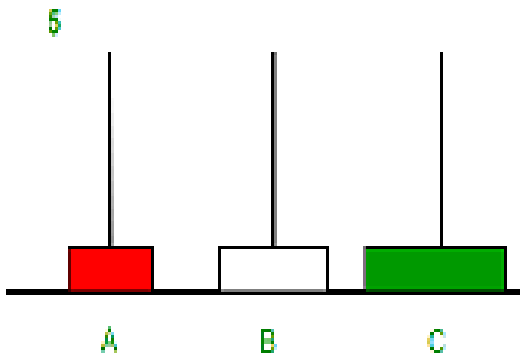

3 Disk
A
B
C

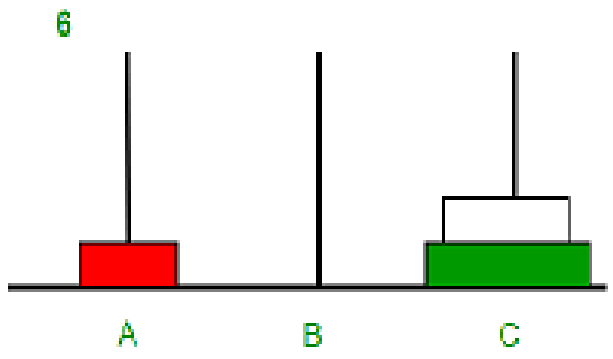

1
A
B
C

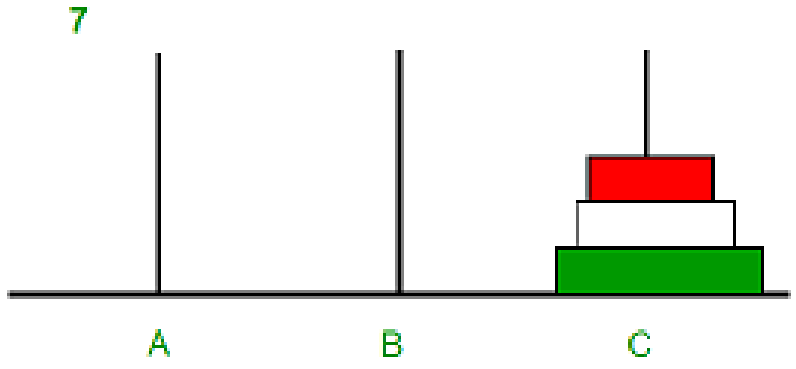

2
A
B
C


3
A
B
C


4
A
B
C


5
A
B
C


6
A
B
C


7
A
B
C

VIT-AP UNIVERSITY

Artificial Intelligence
PLANNING

VIT-AP UNIVERSITY

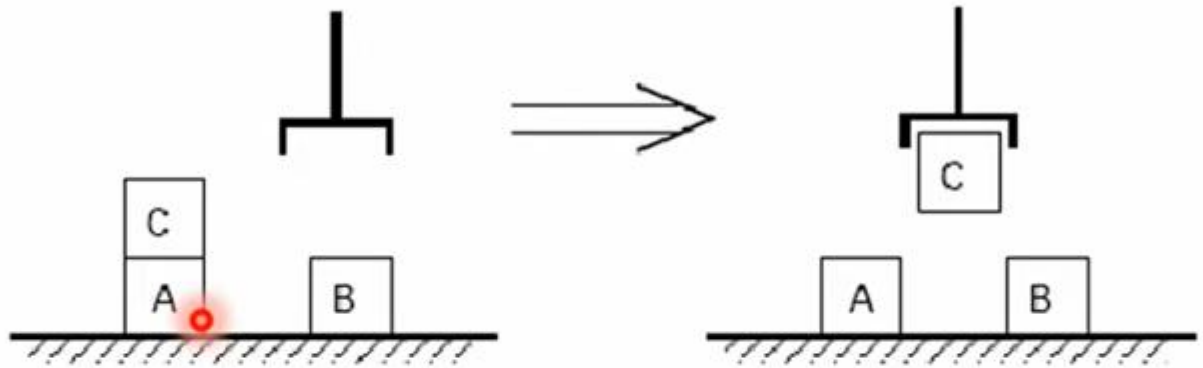
What is Planning in AI?

- The Planning in Artificial Intelligence is about the **decision making tasks performed by the robots or computer programs to achieve a specific goal.**
- The execution of planning is about **choosing a sequence of actions.**
- Planning refers to the **process of computing several steps** of a problem-solving procedure before executing any of them.
- **Planning is arranging a sequence of actions to achieve a goal.**

VIT-AP UNIVERSITY

EXAMPLE-BLOCKS WORLD (BW) PLANNING

- The Blocks world consists of:



→A flat surface such as a table top

→An adequate set of identical blocks which are identified by letters.

→The blocks can be stacked one upon another.

→There is a robot arm that can manipulate the blocks.

→The robot can hold one block at a time and only one block can be moved at a time.

→Any number of blocks can be on the table.

The **actions** it can perform include

- **UNSTACK(A,B)**:
  - remove block A from block B.
  - The arm must be empty
  - block A must have no blocks on top of it.
- **STACK(A,B)**:
  - put block A on block B.
  - The arm must already be holding A
  - the surface of B must be clear.
- **PICKUP(A)**:
  - pickup block A from the table.
  - The arm must be empty and there must be nothing on top of A.
- **PUTDOWN(A)**:
  - put block A on the table.
  - The arm must have been holding block A.

VIT-AP UNIVERSITY

Predicates

The following predicates are needed to perform an operation:

- **ON(A, B)**: Block A is on Block B.
- **ONTABLES(A)**: Block A is on the table.
- **CLEAR(A)**: There is nothing on the top of Block A.
- **HOLDING(A)**: The arm is holding Block A.
- **ARMEMPTY**: The arm is holding nothing.

Logical notation can be used in Block's World.

For eg,

$$[\exists x : \text{HOLDING}(x)] \rightarrow \neg \text{ARMEMPTY}$$
$$\forall x : \text{ONTABLE}(x) \rightarrow \neg \exists y : \text{ON}(x, y)$$
$$\forall x : [\neg \exists y : \text{ON}(y, x)] \rightarrow \text{CLEAR}(x)$$

1. If the arm is holding anything, then it is not empty.
2. If a block is on the table, then it is not also on another block.
3. Any block with no blocks on it is clear.

VIT-AP UNIVERSITY

COMPONENTS OF A PLANNING SYSTEM

In problem solving systems, it is necessary to perform each of the following functions:

- **Choose the best rule based upon heuristic information**
- **Apply this rule to create a new state**
- **Detect when a solution has been found**
- **Detect dead ends so that they can be avoided.**
- **Detect when almost correct solution has been found and employ special techniques to make it totally correct.**

STRIPS

- STRIPS stands for **"STanford Research Institute Problem Solver,"** was the planner used in Shakey, one of the first robots built using AI technology ,which is an action-centric representation ,for each action , specifies the effect of an action.

A STRIPS planning problem specifies:
1) an initial state *S*
2) a goal *G*
3) a set of STRIPS actions

- The STRIPS representation for an action consists of three lists,

  - Pre_Cond list contains predicates which have to be true before operation.
  - ADD list contains those predicates which will be true after operation
  - DELETE list contain those predicates which are no longer true after operation

What is STRIPS?

VIT-AP UNIVERSITY

- The STandford Research Institute Problem Solver (**STRIPS**), is an automated planning technique used to find a goal, by executing a domain from the initial state of domain.

- With STRIPS, we can first describe the world, (Initial state and goal state) by providing **objects, actions, preconditions, and effects**

- To describe the world, we used two categories of terms
  - States – initial state and goal state
  - Action schema – objects, actions, preconditions, and effects

VIT-AP UNIVERSITY

- Once the world is described, then provide a problem set.
- A problem consists of an initial state and a goal condition.
- STRIPS can then search all possible states, starting from the initial one, executing various actions, until it reaches the goal.

VIT-AP UNIVERSITY

Planning Domain Definition Language (PDDL).

- A common language for writing STRIPS domain and problem sets, is the Planning Domain Definition Language (**PDDL**).
- In PDDL most of the codes are English words, so that it can be clearly read and well understood.
- It's a relatively easy approach to writing simple AI planning problems.

STRIPS - States, Goals and Actions

VIT-AP UNIVERSITY

- States: conjunctions of ground, function-free, and positive literals, such as At(Home) ^ Have(Banana)
- To describe states, Closed-world assumption is used (the world model contains everything, the agent needs to know: there can be no surprise)

- Goals: conjunctions of literals, may contain variables (existential), goal may represent more than one state
  - E.g. At(Home) ^ ¬Have(Bananas)
  - E.g. At(x) ^ Sells(x, Bananas)

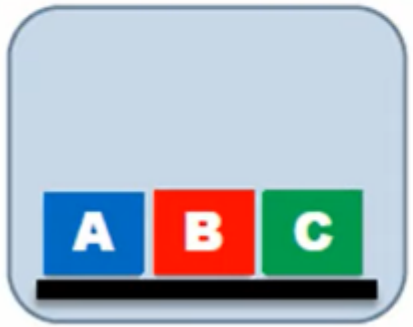
- Actions: preconditions that must hold before execution and the effects after execution

STRIPS Action Schema

VIT-AP UNIVERSITY

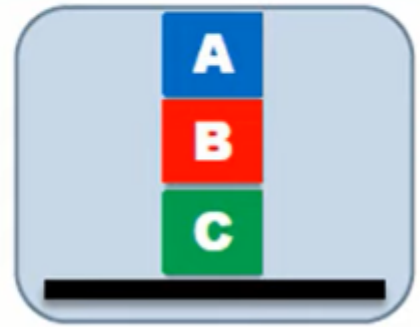
- An action schema includes:
  - Action name & parameter list (variables)
  - Precondition: a conjunction of function-free positive literals. The action variables must also appear in precondition.
  - Effect: a conjunction of function-free literals (positive or negative)
- Add-list: positive literals
- Delete-list: negative literals
- Example:
  - Action: Buy (x)
  - Precondition: At (p), Sells (p, x)
  - Effect: Have(x)

STRIPS planning



On(A,Table)
On(B,Table)
On(C,Table)
Clear(A)
Clear(B)
Clear(C)

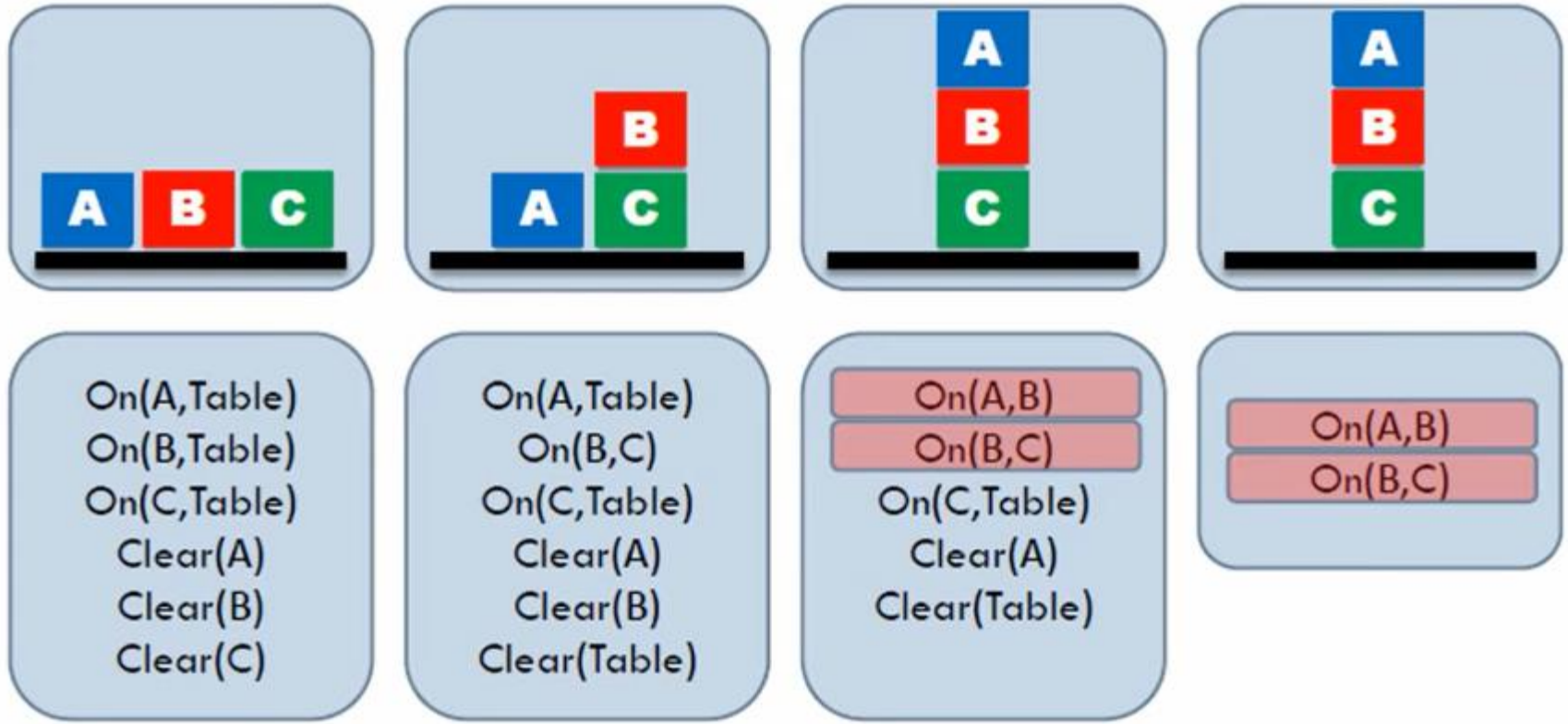
s_0



On(A,B)
On(B,C)

g

STRIPS planning



Air Cargo Transport

VIT-AP UNIVERSITY

- An air cargo transport problem involving loading cargo onto planes, and unloading cargo off of planes, and flying the planes from one place to another place.
- The problem can be defined with three actions:
  - Load, Unload, and Fly.
- The actions affect two predicates:
  - In(c, p) means that cargo c is inside plane p, and At (x, a) means that object x (either plane or cargo) is at airport a.
- The following plan is a solution to the problem:

[Load(C1, P1, SFO), Fly(P1, SFO, JFK)
Load(C2, P2, JFK), Fly(P2, JFK, SFO)]

A STRIPS problem involving transportation of air cargo between airports.

VIT-AP UNIVERSITY

$$
\begin{aligned}
&Init(At(C_1, SFO) \land At(C_2, JFK) \land At(P_1, SFO) \land At(P_2, JFK) \\
&\quad \land Cargo(C_1) \land Cargo(C_2) \land Plane(P_1) \land Plane(P_2) \\
&\quad \land Airport(JFK) \land Airport(SFO)) \\
&Goal(At(C_1, JFK) \land At(C_2, SFO)) \\
&Action(Load(c, p, a), \\
&\quad \text{PRECOND: } At(c, a) \land At(p, a) \land Cargo(c) \land Plane(p) \land Airport(a) \\
&\quad \text{EFFECT: } \neg At(c, a) \land In(c, p)) \\
&Action(Unload(c, p, a), \\
&\quad \text{PRECOND: } In(c, p) \land At(p, a) \land Cargo(c) \land Plane(p) \land Airport(a) \\
&\quad \text{EFFECT: } At(c, a) \land \neg In(c, p)) \\
&Action(Fly(p, from, to), \\
&\quad \text{PRECOND: } At(p, from) \land Plane(p) \land Airport(from) \land Airport(to) \\
&\quad \text{EFFECT: } \neg At(p, from) \land At(p, to))
\end{aligned}
$$

VIT-AP UNIVERSITY

- Ex: given a chair and a table, the goal is to have them match—have the same color. In the initial state we have two cans of paint, but the colors of the paint and the furniture are unknown. Only the table is initially in the agent's field of view:

$$Init(Object(Table) \land Object(Chair) \land Can(C_1) \land Can(C_2) \land InView(Table))$$
$$Goal(Color(Chair, c) \land Color(Table, c))$$

- two actions: removing the lid from a paint can and painting an object using the paint from an open can

VIT-AP UNIVERSITY

- Action Schema

$$
\begin{aligned}
&Action(RemoveLid(can),\\
&\quad \textsc{Precond}: Can(can)\\
&\quad \textsc{Effect}: Open(can))\\
&Action(Paint(x, can),\\
&\quad \textsc{Precond}: Object(x) \wedge Can(can) \wedge Color(can, c) \wedge Open(can)\\
&\quad \textsc{Effect}: Color(x, c))
\end{aligned}
$$

- Percept Schema

$$
\begin{aligned}
&Percept(Color(x, c),\\
&\quad \textsc{Precond}: Object(x) \wedge InView(x)\\
&Percept(Color(can, c),\\
&\quad \textsc{Precond}: Can(can) \wedge InView(can) \wedge Open(can)
\end{aligned}
$$

Solution with Planning

VIT-AP UNIVERSITY

$$
\begin{aligned}
&[LookAt(Table), LookAt(Chair),\\
&\quad \textbf{if } Color(Table, c) \wedge Color(Chair, c) \textbf{ then } NoOp\\
&\qquad \textbf{else } [RemoveLid(Can_1), LookAt(Can_1), RemoveLid(Can_2), LookAt(Can_2),\\
&\qquad\quad \textbf{if } Color(Table, c) \wedge Color(can, c) \textbf{ then } Paint(Chair, can)\\
&\qquad\quad \textbf{else if } Color(Chair, c) \wedge Color(can, c) \textbf{ then } Paint(Table, can)\\
&\qquad\quad \textbf{else } [Paint(Chair, Can_1), Paint(Table, Can_1)]]]
\end{aligned}
$$

K-STRIPS

VIT-AP UNIVERSITY

- We are familiar with the use of connectives ∧ and V in logics. Thinking of these connectives as operators that construct more complex formulas from simpler components. Here, we want to construct a formula whose intended meaning is that a certain agent knows a certain proposition.

- The components consist of a term denoting the agent and a formula denoting a proposition that the agent knows. To accomplish this, modal operator K is introduced.

- For example, to say that Robot (name of agent) know that block A is on block B, then write,
  K(Robot, On(A,B))

K-STRIPS

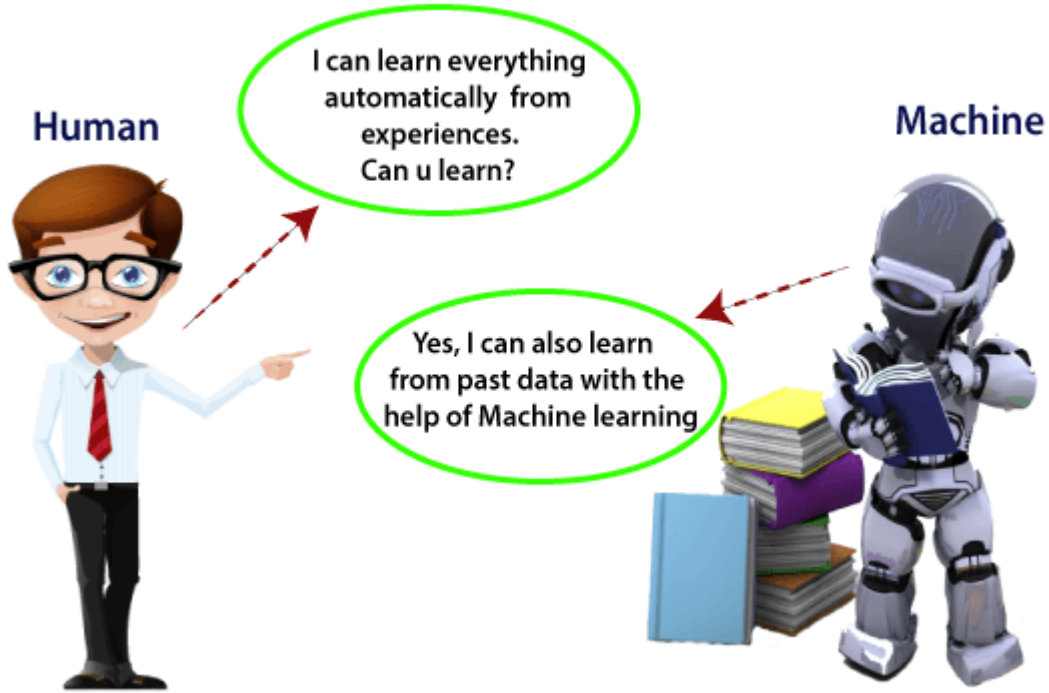
VIT-AP UNIVERSITY

- The words “knows” and “belief” is different in meaning. That means an agent can believe a false proposition, but it cannot know anything that is false.
- Some examples,
- K(Agent1, K(Agent2, On(A,B))], means Agent1 knows that Agent2 knows that A is on B.
- K(Agent1, On(A,B)) V K(Agent1, On(A,C)) means that either Agent1 knows that A is on B or it knows that A is on C.
- K(Agent1, On(A,B)) V K(Agent1, ¬On(A,B)) means that either Agent1 knows whether or not A is on B.

Machine Learning

VIT-AP UNIVERSITY

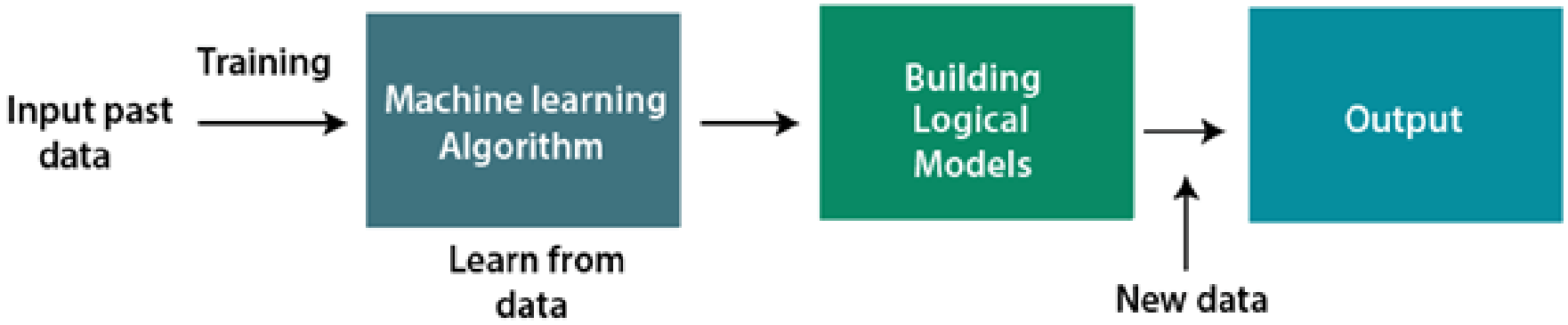
Machine Learning



> "Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed."

How does Machine Learning work

A Machine Learning system **learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it**. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

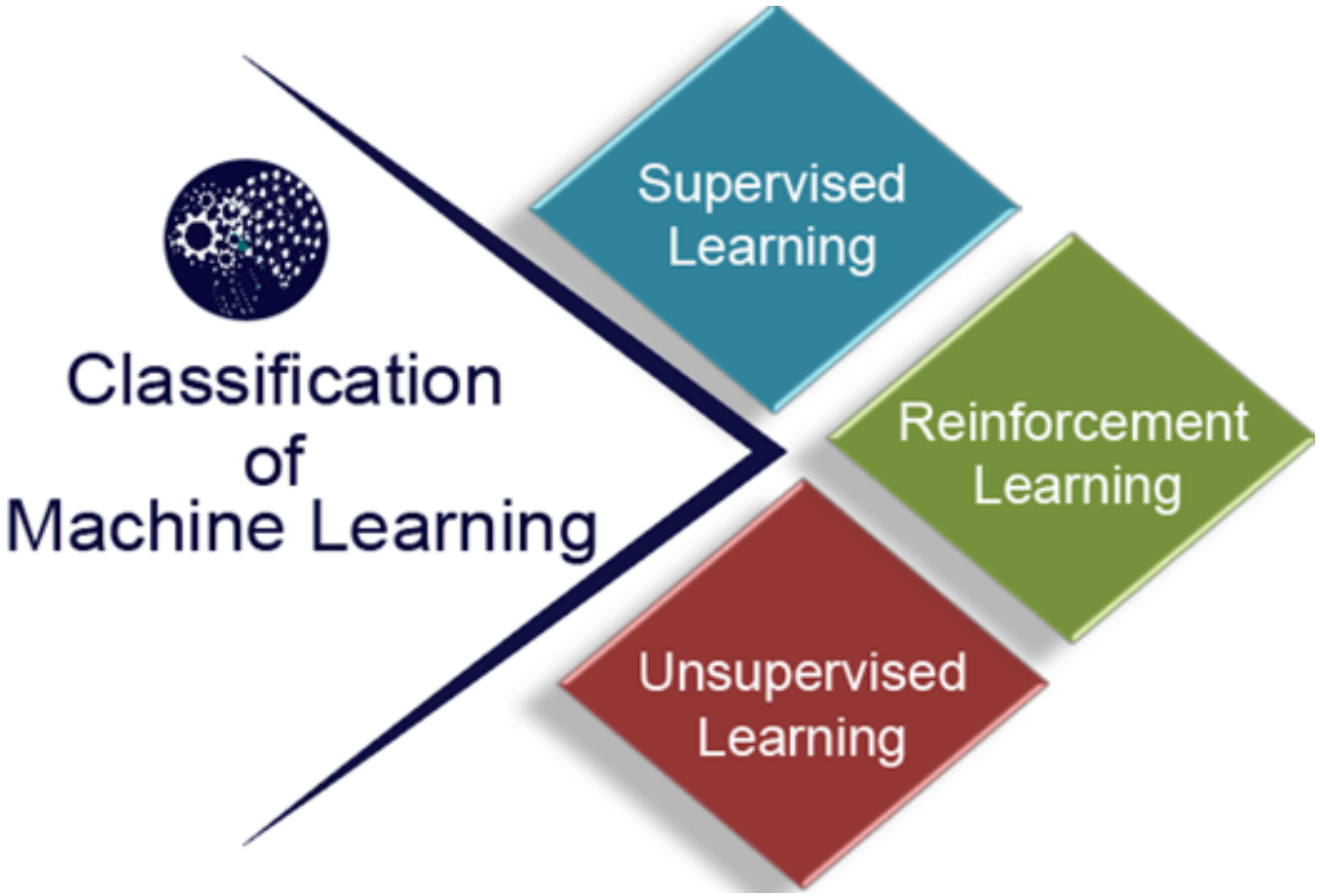


Classification of Machine Learning

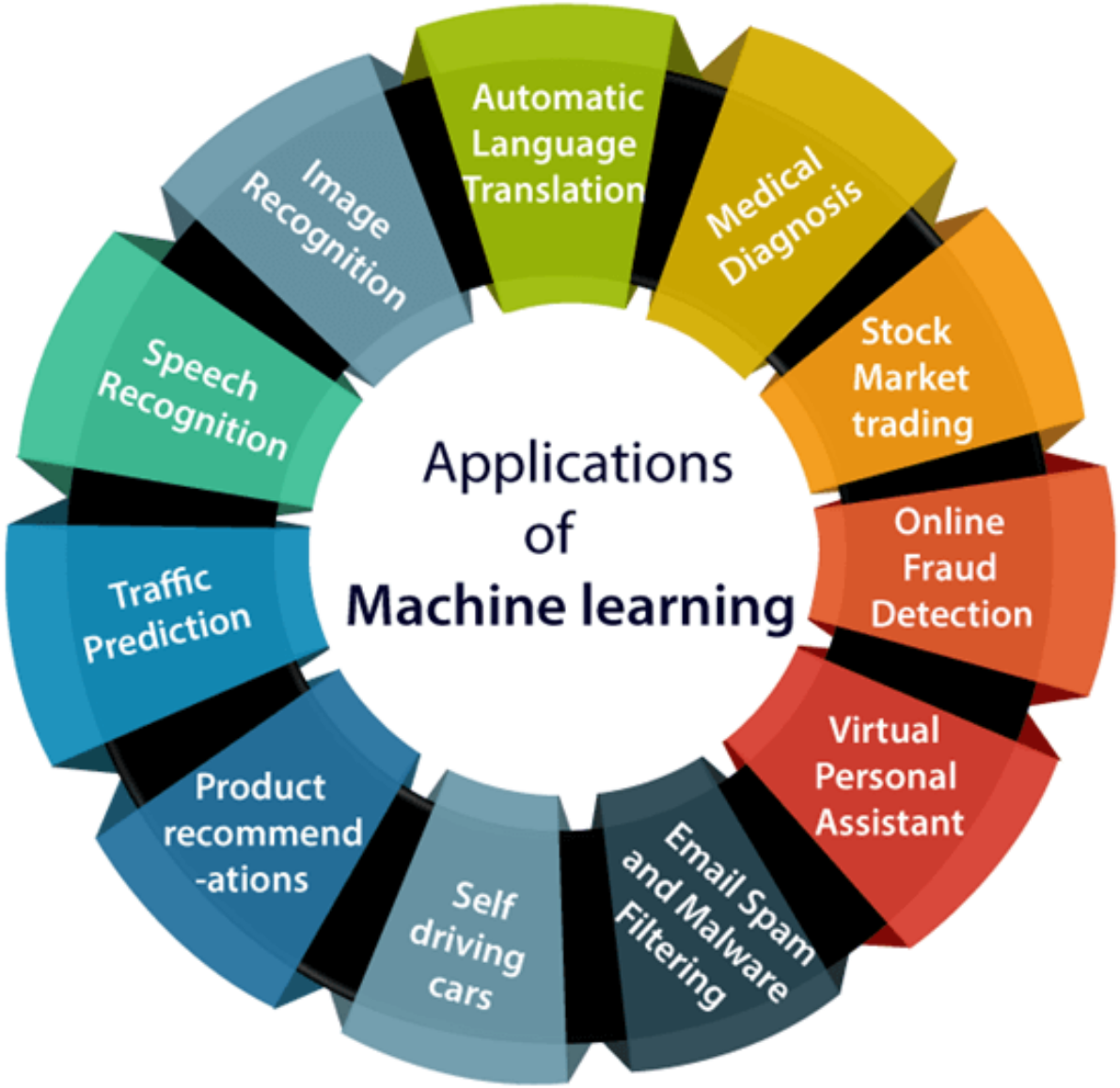
VIT-AP UNIVERSITY

At a broad level, machine learning can be classified into three types:

1. **Supervised learning**
2. **Unsupervised learning**
3. **Reinforcement learning**

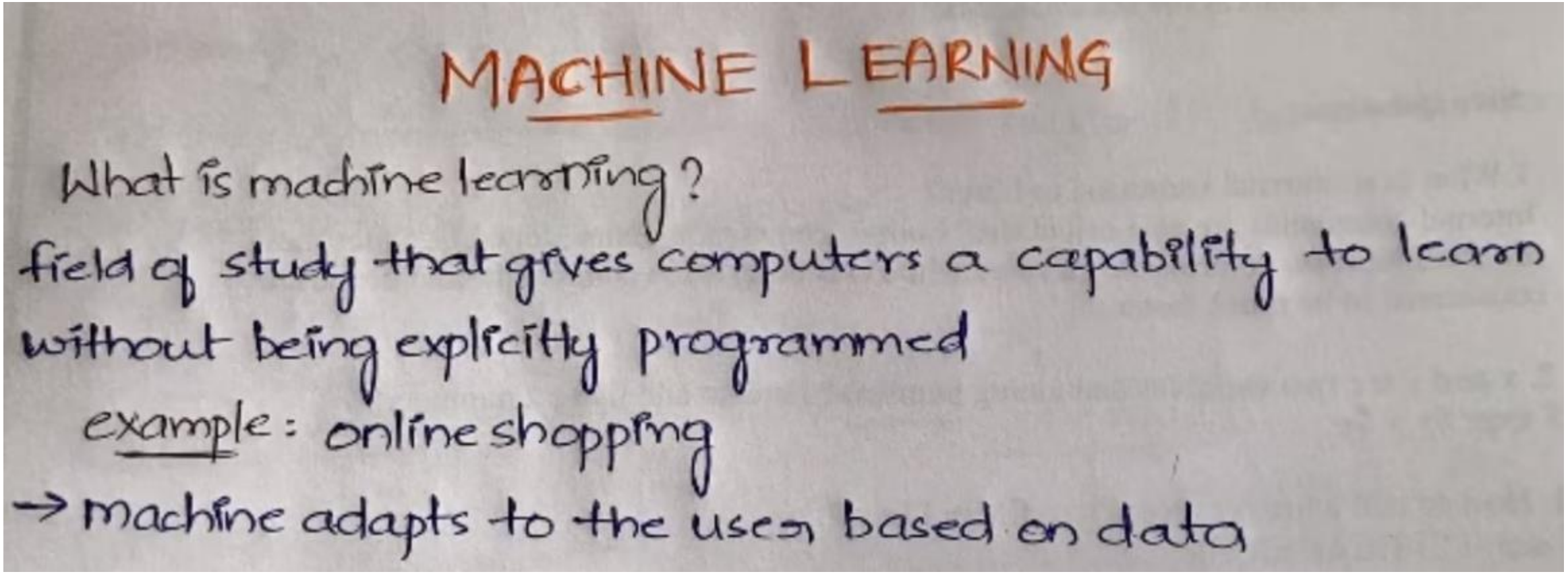


Applications of Machine learning



Machine Learning

VIT-AP UNIVERSITY



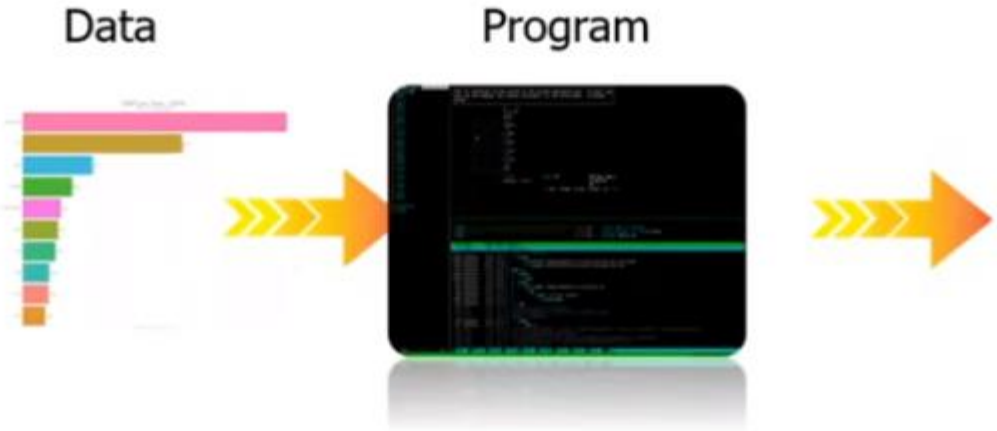
VIT-AP UNIVERSITY

What is Machine Learning ?

Software Used – 3Ds Max

VIT-AP UNIVERSITY

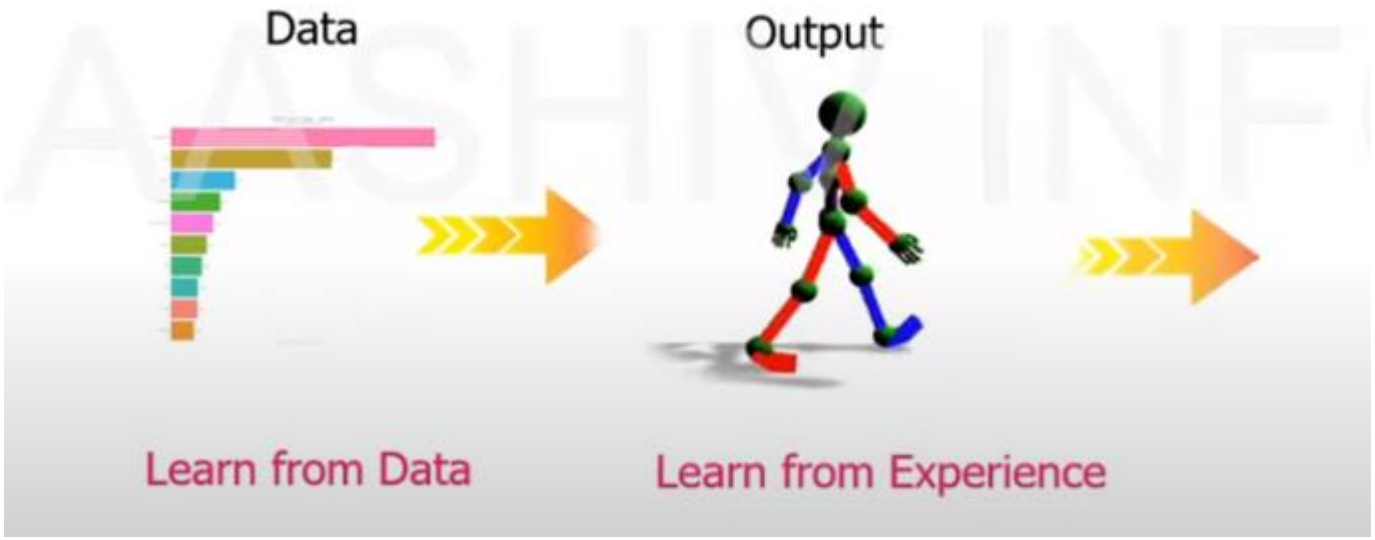
Traditional Programming

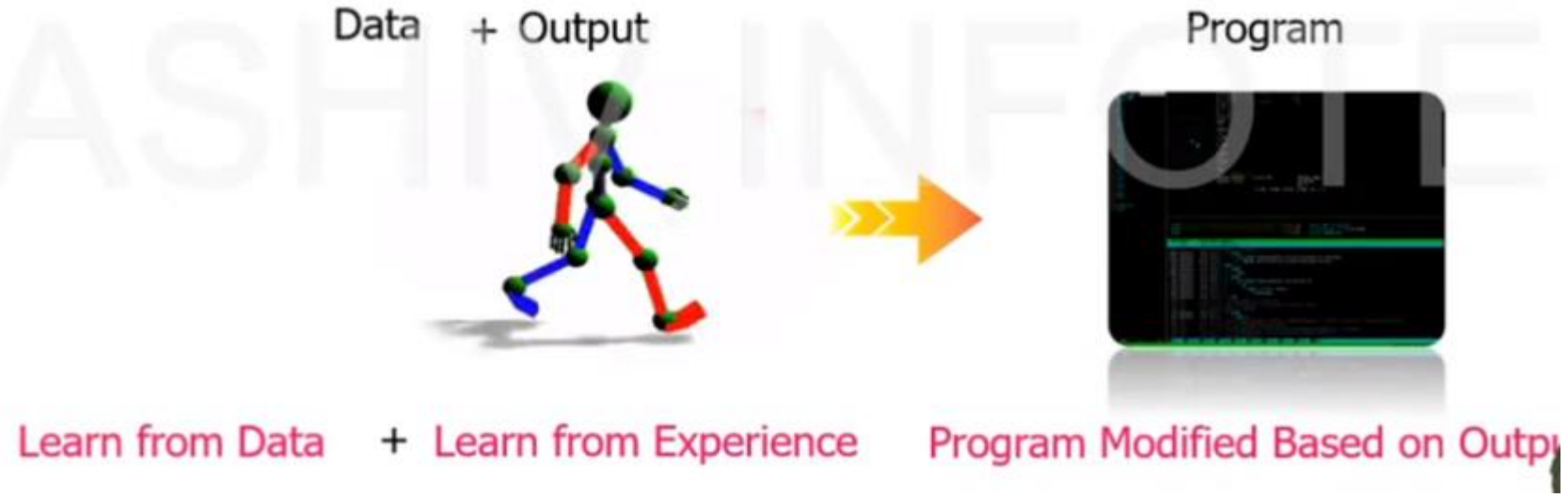


Traditional Programming

VIT-AP UNIVERSITY

Machine Learning Programming

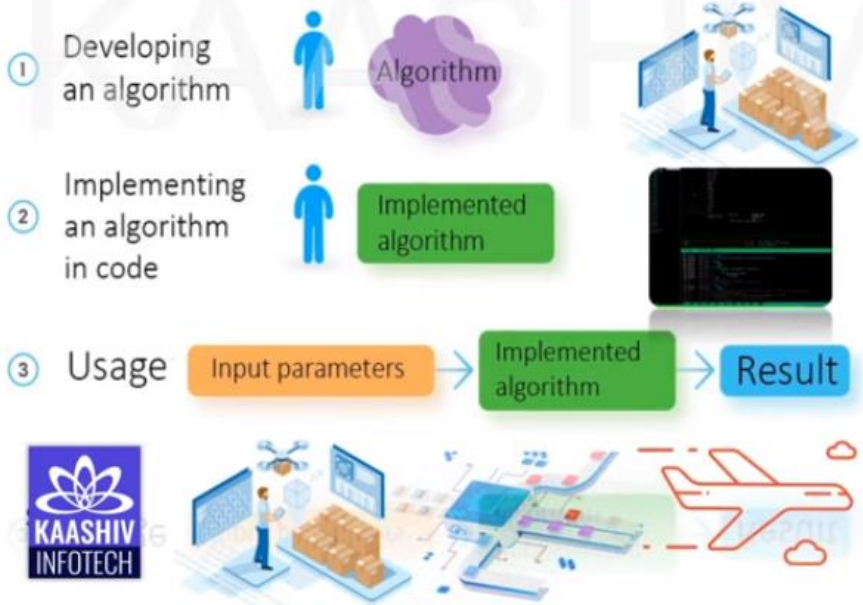




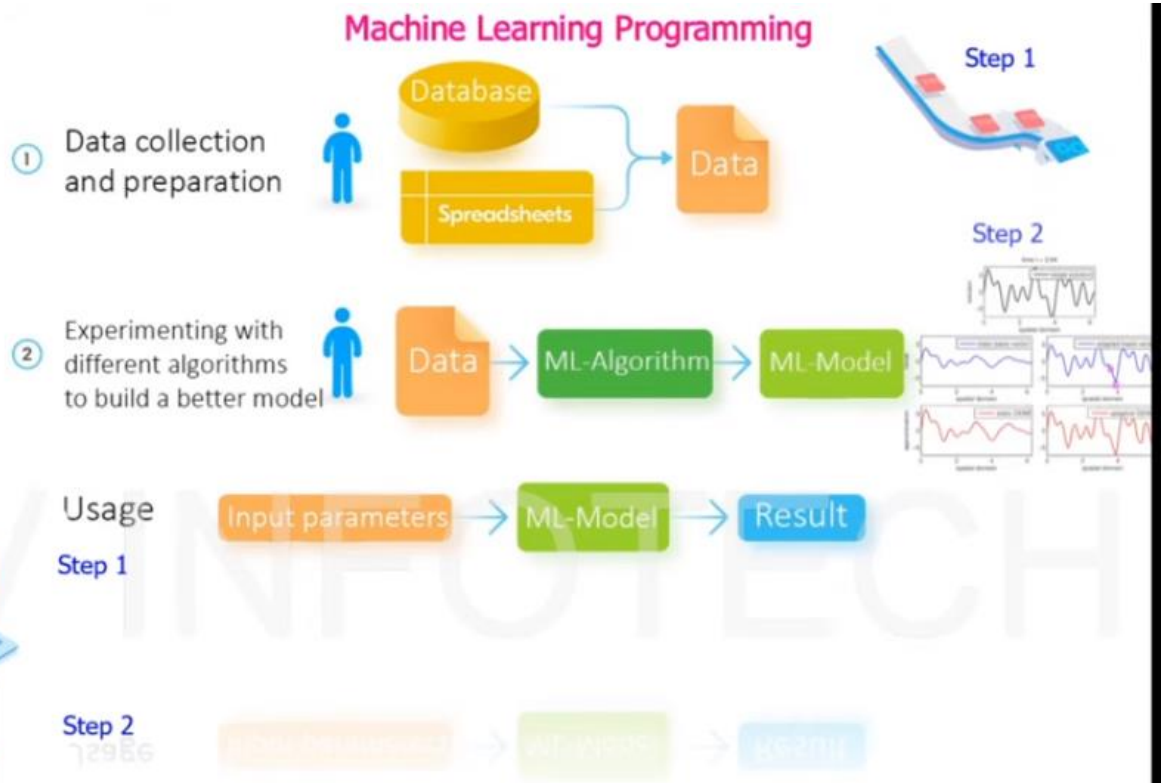
How Machine Learning Works ?

Software Used – 3Ds Max

Traditional Programming

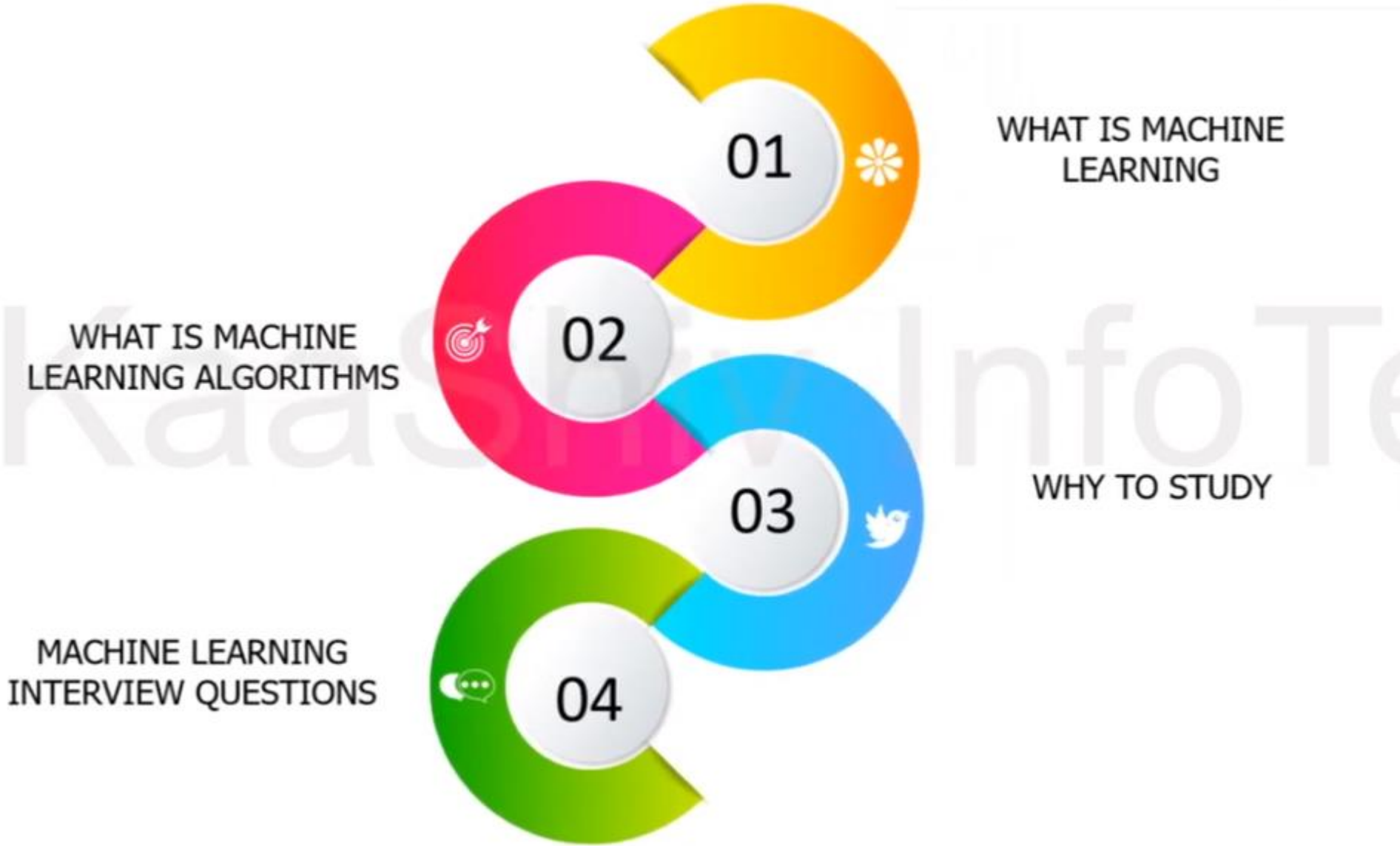


Machine Learning Programming

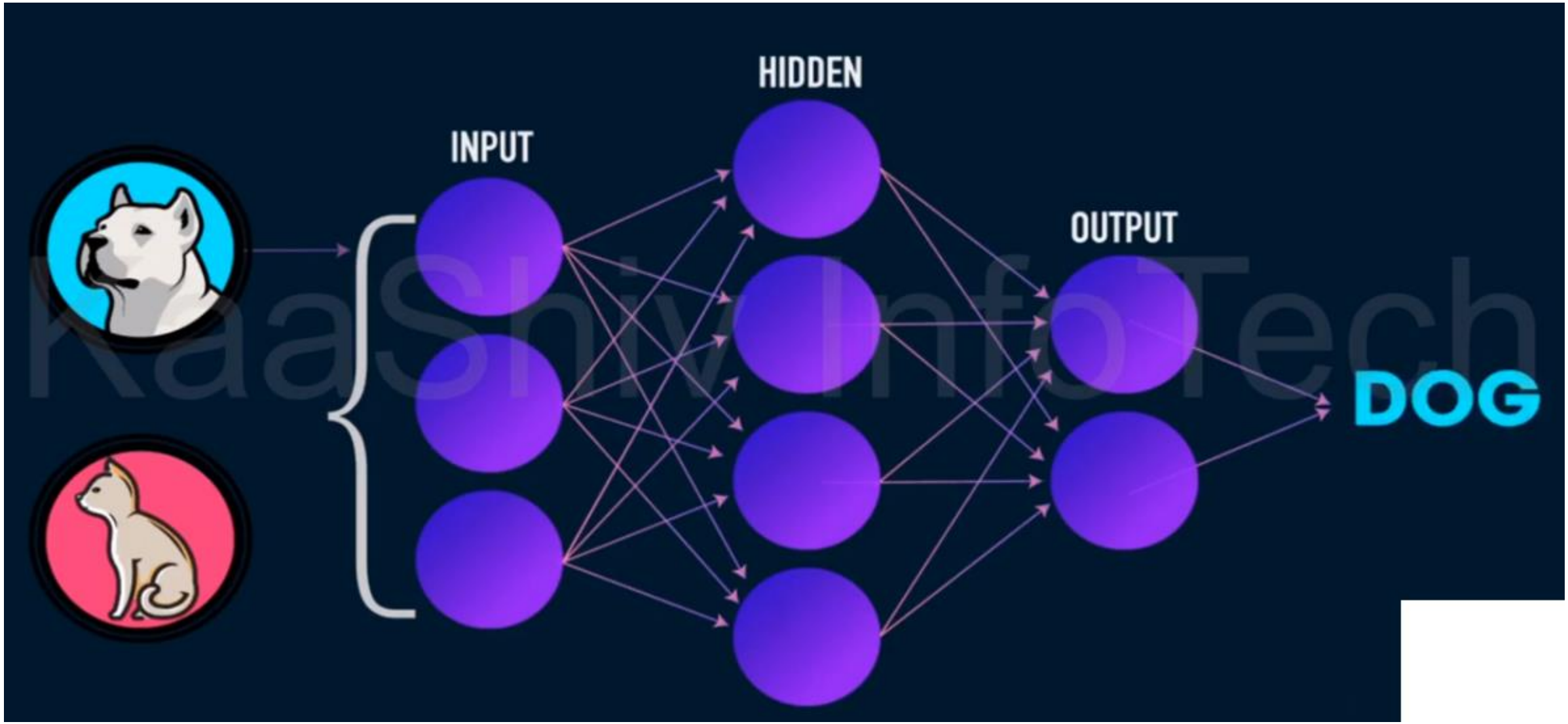


VIT-AP UNIVERSITY

VIT-AP UNIVERSITY



VIT-AP UNIVERSITY



NE LEARNING
ORITHMS

SUPERVISED LEARNING

- Pre – Categorized Data
- Learning in the presence of an Expert / Teacher
- Data is labelled with a class or variable

UNSUPERVISED LEARNING

- Unlabeled Data
- No Knowledge of Output class or value / Self guided learning Algorithm
- Data is unlabeled and value is unknown

REINFORCEMENT LEARNING

- No Predefined Data
- An Agent interacts with its environment by performing actions & Learning from errors

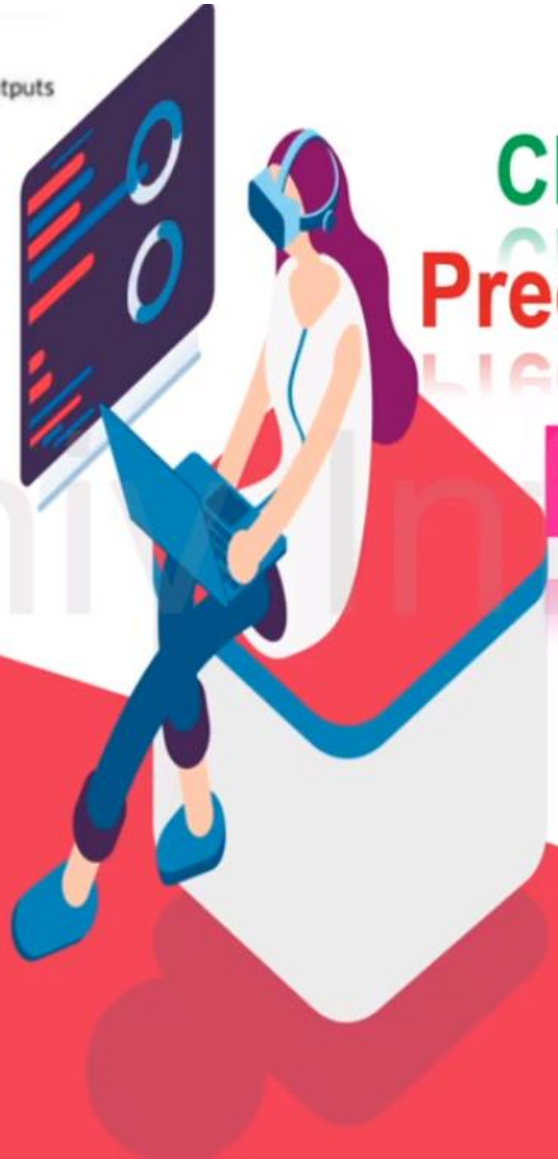
Learning fro
mistakes
(Playing Gam

VIT-AP UNIVERSITY

VIT-AP UNIVERSITY

SUPERVISED LEARNING

- Pre – Categorized Data
- Learning in the presence of an Expert / Teacher
- Data is labelled with a class or variable

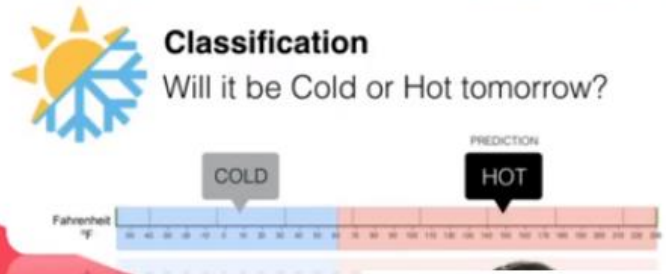


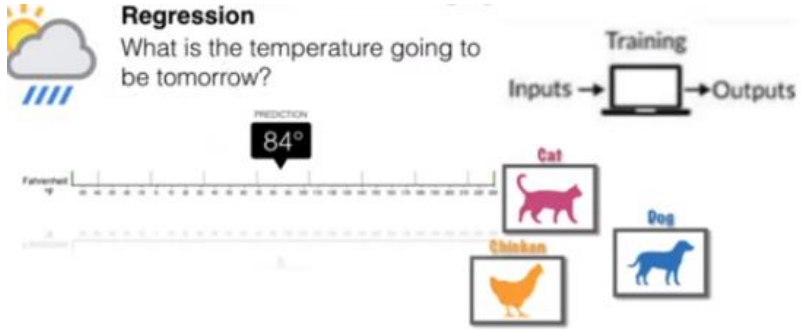
SUPERVISED LEARNING

CLASSIFICATION

Predicting Students

Pass OR Fail





SUPERVISED LEARNING

REGRESSION

Predicting Students

Marks

OR

Percentage

SUPERVISED LEARNING

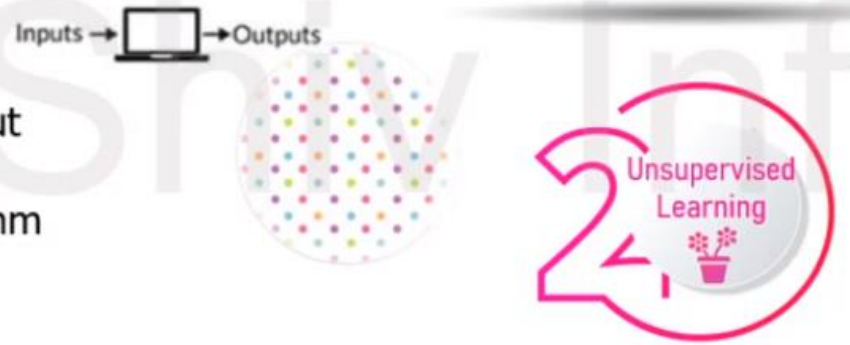
- Pre – Categorized Data
- Learning in the presence of an Expert / Teacher
- Data is labelled with a class or variable

VIT-AP UNIVERSITY

SUPERVISED LEARNING

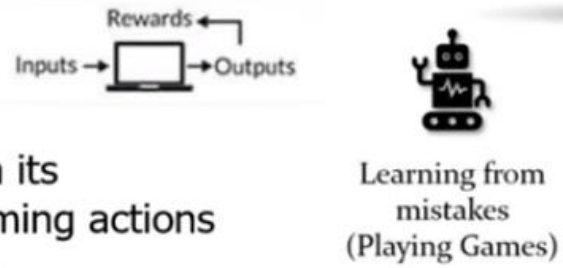
- Pre – Categorized Data
- Learning in the presence of an Expert / Teacher
- Data is labelled with a class or variable

UNSUPERVISED LEARNING



- Unlabeled Data
- No Knowledge of Output class or value / Self guided learning Algorithm
- Data is unlabeled and value is unknown

REINFORCEMENT LEARNING



- No Predefined Data
- An Agent interacts with its environment by performing actions & Learning from errors

What is Adaptive ML?

VIT-AP UNIVERSITY

- Virtually every business relies on data to survive.
- Sales data gives you insights into your business's performance.
- Customer data tells you more about your target audience and their behaviors.
- Marketing data helps you understand how to improve brand awareness and expand your reach.
- Competitor data, transaction data, financial data, employee data – the list goes on.
- Without it all, you wouldn't be able to make informed decisions that could take your company forw

- But it's one thing collecting data, and quite another analyzing it and transforming it into valuable, actionable information. And that's precisely where machine learning comes in.
- Most commonly, businesses rely on traditional ML to handle data collection, analysis, and predictions, but adaptive ML has started taking the spotlight.

Why Traditional Machine Learning Doesn't Cut It

VIT-AP UNIVERSITY

- Traditional ML involves only two primary pipelines – one for training (responsible for data collection) and the other for making predictions (responsible for data analysis).
- Although you'll commonly see AI products being marketed as "always learning, always evolving," that's typically not the case if the product relies on this traditional batch-based ML model.
- Traditional machine learning is static; it depends on parameters that don't change, making it great for horizontal scalability but causing problems in dynamic industries where data changes quickly.
- Since there are only two pipelines for data collection and analysis, and since traditional ML models rely on past data to generate new predictions, you can never have true, real-time insights critical in industries such as e-commerce, for example, where trends are constantly changing.

Overcoming the traditional ML flaws

VIT-AP UNIVERSITY

To overcome the inherent flaws of traditional ML models, developers typically commit to one of two approaches:

- Manually training for new data
- Scheduling automatic training for new data

- Manual training for new data is a time-consuming process that doesn't deliver much better results, so most developers opt for the second option.

The Adaptive ML/ Online Machine Learning Advantage

VIT-AP UNIVERSITY

- **Adaptive machine learning is a more advanced solution that takes real-time data collection and analysis seriously. As its name would suggest, it easily adapts to new information and provides insights almost instantaneously.**

- Instead of having a two-channel or two-pipeline approach like traditional ML, adaptive ML relies on a single channel. As opposed to batch learning, adaptive learning collects and analyzes data in sequential order, not all at once.

- This enables adaptive ML models to monitor and learn from the changes in both input and output values; it allows the model to adapt its data collection, grouping, and analysis methods based on new information.

Adaptive Learning

VIT-AP UNIVERSITY

- Traditional Machine learning fails if
  - Changing the system's operational surrounding
  - Changing the system's input
  - Changing the desired outcomes or results

- In today's pace of living, large amounts of data need to be transferred quickly, but sometimes they need to change in the same timely manner.
-
- Adaptive ML, as its name says, can do something that traditional ML can't. It can quickly adapt to new information and gain insight into how important that new information is.

Adaptive Learning

VIT-AP UNIVERSITY

- Adaptive → the system is learning and updating as long as the new information is provided. This single-channeled system follows up on every feedback provided to make future predictions and outcomes even better.
- Differences between Adaptive ML and Traditional ML:
  - Firstly, the main difference is that traditional ML is a system with two channels. Consequently, all data is divided into two parts, as previously mentioned. Even though traditional ML finds a way to deal with this, the results can take a long time and often even become outdated by the time they arrive.
  - Therefore, the adaptive ML works on a single channel, providing a more efficient way of functioning, relating to the resources used and the speed of solutions. Thereby, this system proves to be a more sustainable and better option overall.

Adaptive Learning

VIT-AP UNIVERSITY

- Secondly, as traditional ML goes through fewer data in more time, it's mainly based on static and permanent data. It takes quite a lot of time to change the behavior in the system, so numerous urgent and important matters are missed.
- That's why adaptive ML isn't based on any permanent data but on the ability to continuously adapt and change the behavior when necessary, making sure that the system doesn't operate on any outdated data.
- Finally, adaptive ML has the unique ability to learn from the past. In that way, it resembles feeding someone – the more information you feed to the system, the bigger and smarter it gets. It even learns from previous mistakes and lowers the chances of repeating them. So, the longer they operate, the more accurate they get.

VIT-AP UNIVERSITY

The Pros and Cons of Adaptive Machine Learning

Adaptive machine learning brings several unique benefits that could be useful across industries. Its main pros include:

- Robustness
- Efficiency
- Agility
- Accuracy
- Sustainability

Applications of Adaptive Machine Learning

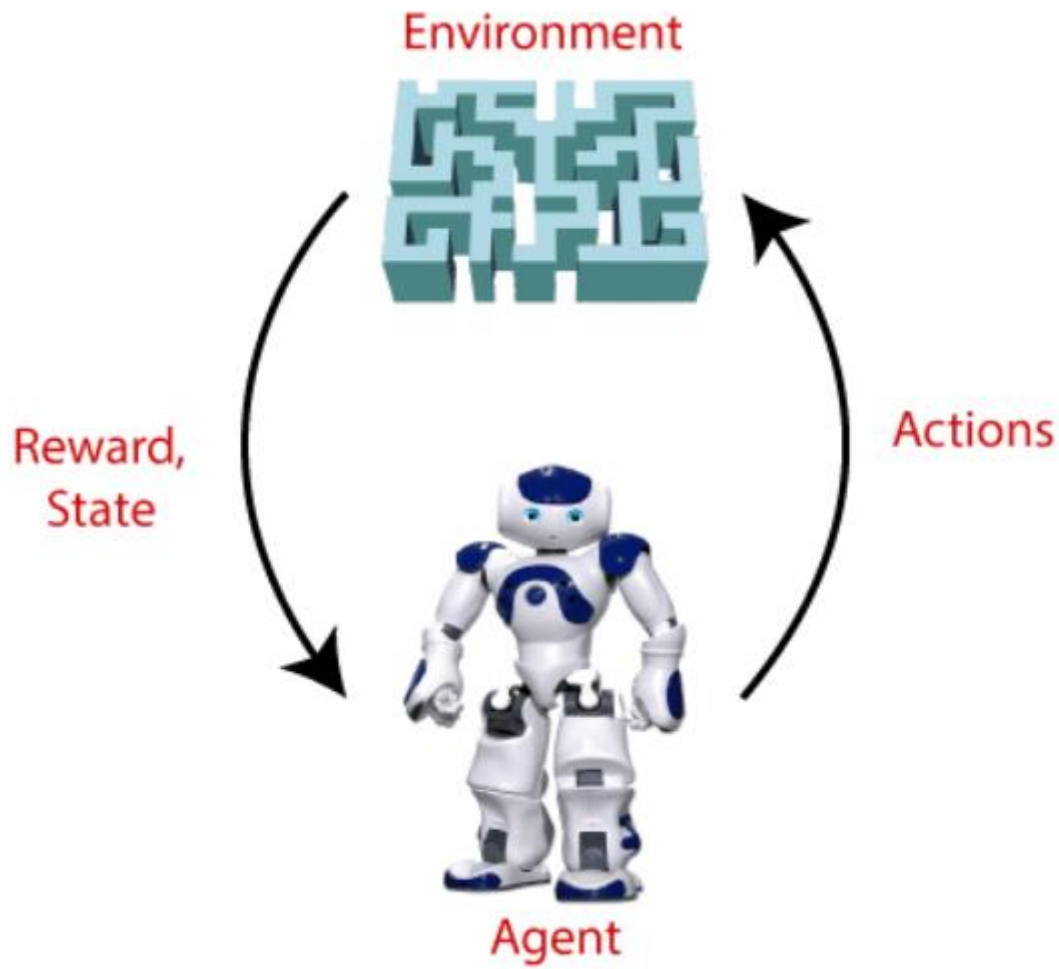
Considering its agility, precision, and real-time capabilities, adaptive ML can be valuable across industries and niches:

- Financial sector – for detecting fraudulent transactions, predicting trends, automating trading systems;
- Manufacturing sector – for predicting system malfunctions and solving supply chain issues;
- Healthcare – for providing quick, accurate, and affordable diagnoses and alerting of problems before they arise;
- Marketing – for perfecting marketing campaigns based on real-time insights;
- E-commerce – for predicting trends and devising effective strategies;

As more and more industries start relying on adaptive ML technology, it will become evident just how powerful these models can be.

Reinforcement Learning

- Reinforcement Learning is a feedback-based Machine learning technique in which an agent learns to behave in an environment by performing the actions and seeing the results of actions. For each good action, the agent gets positive feedback, and for each bad action, the agent gets negative feedback or penalty.
- In Reinforcement Learning, the agent learns automatically using feedbacks without any labeled data, unlike supervised learning.
- Since there is no labeled data, so the agent is bound to learn by its experience only.


Environment
Reward,
State
Actions
Agent

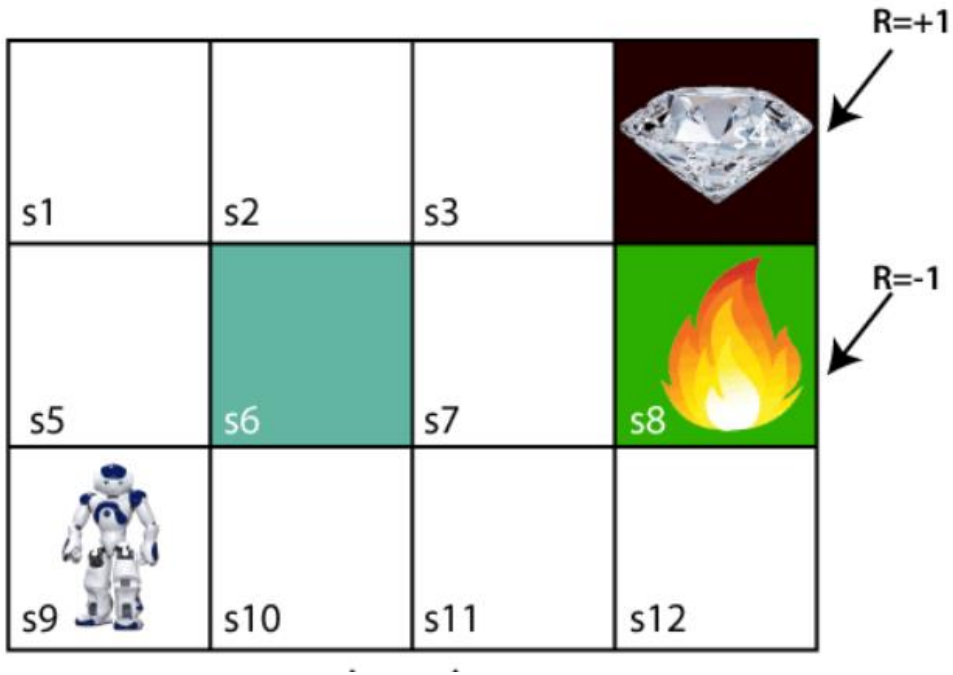
How does Reinforcement Learning Work?

VIT-AP UNIVERSITY

To understand the working process of the RL, we need to consider two main things:

- **Environment:** It can be anything such as a room, maze, football ground, etc.
- **Agent:** An intelligent agent such as AI robot.

Let's take an example of a maze environment that the agent needs to explore. Consider the below image:



In the above image, the agent is at the very first block of the maze. The maze is consisting of an S_6 block, which is a **wall**, S_8 a **fire pit**, and S_4 a **diamond block.**

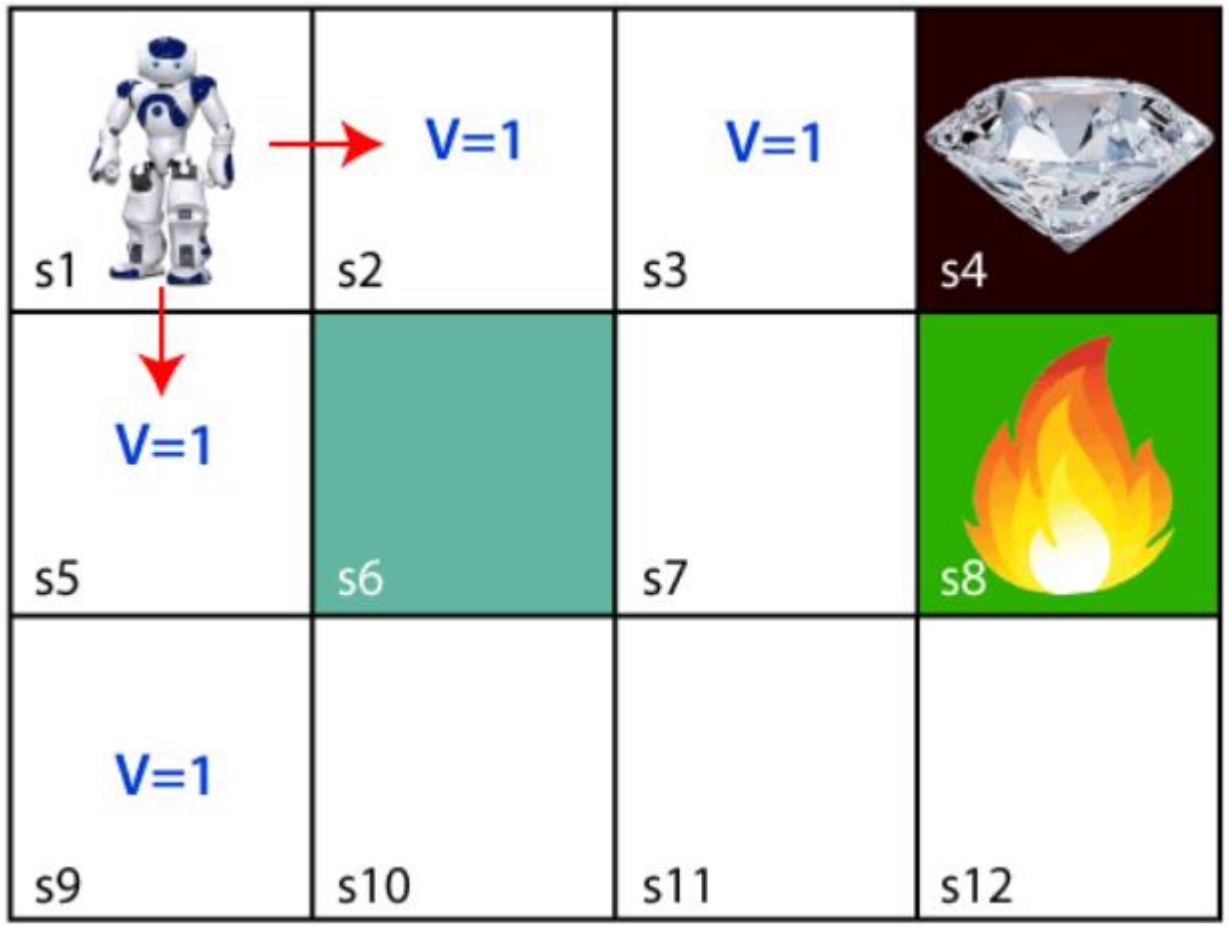
The agent cannot cross the S_6 block, as it is a solid wall. If the agent reaches the S_4 block, then get the **+1 reward;** if it reaches the fire pit, then gets **-1 reward point**. It can take four actions**: move up, move down, move left, and move right.**

The agent can take any path to reach to the final point, but he needs to make it in possible fewer steps. Suppose the agent considers the path **S9-S5-S1-S2-S3**, so he will get the +1-reward point.

The agent will try to remember the preceding steps that it has taken to reach the final step. To memorize the steps, it assigns 1 value to each previous step. Consider the below step:

V=1 s1	V=1 s2	V=1 s3	 s4
V=1 s5	s6	s7	 s8
 V=1 s9	s10	s11	s12

Now, the agent has successfully stored the previous steps assigning the 1 value to each previous block. But what will the agent do if he starts moving from the block, which has 1 value block on both sides? Consider the below diagram:



The Bellman Equation

VIT-AP UNIVERSITY

The Bellman equation was introduced by the Mathematician **Richard Ernest Bellman in the year 1953**, and hence it is called as a Bellman equation. It is associated with dynamic programming and used to calculate the values of a decision problem at a certain point by including the values of previous states.

It is a way of calculating the value functions in dynamic programming or environment that leads to modern reinforcement learning.

The key-elements used in Bellman equations are:

Bellman Equation

V=0.81 s1	V=0.9 s2	V=1 s3	 s4
V=0.73 s5	s6	V=0.9 s7	 s8
V=0.66 s9	V=0.73 s10	V=0.81 s11	V=0.73 s12

Reinforcement Learning

VIT-AP UNIVERSITY

The Bellman Equation

- The key-elements used in Bellman equations are:
  - Action performed by the agent is referred to as "a"
  - State occurred by performing the action is "s."
  - The reward/feedback obtained for each good and bad action is "R."
  - A discount factor is Gamma "γ."

The Bellman Equation

VIT-AP UNIVERSITY

- The Bellman equation can be written as:

$$V(s) = \max[R(s,a) + \gamma V(s')]$$

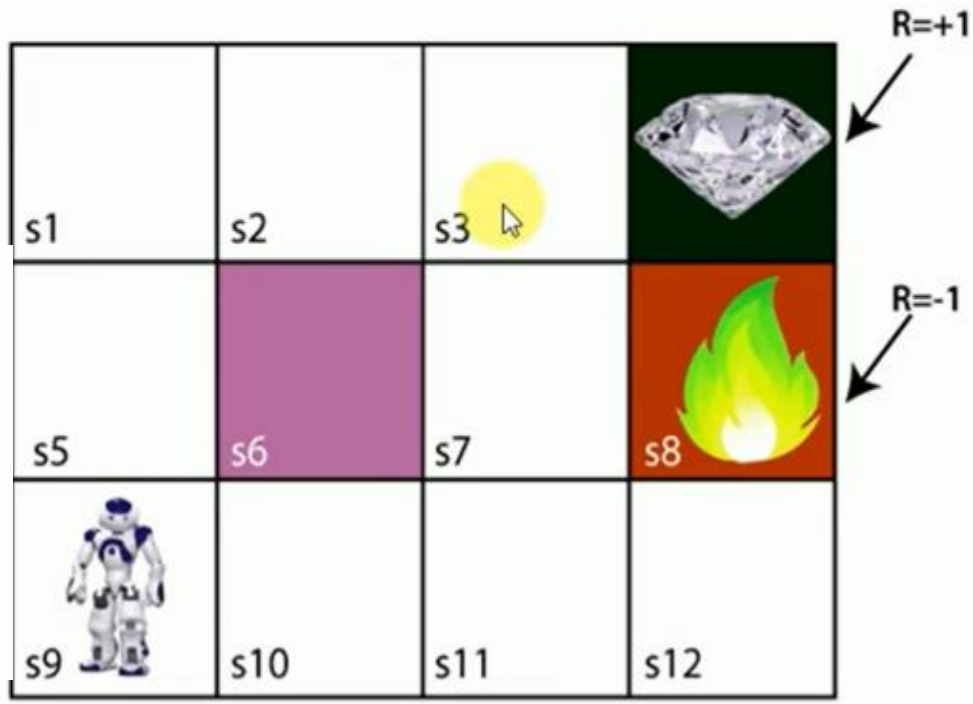
- Where,
  - $V(s) = value\ calculated\ at\ a\ particular\ point.$
  - $R(s,a) = Reward\ at\ a\ particular\ state\ s\ by\ performing\ an\ action\ a.$
  - $\gamma = Discount\ factor$
  - $V(s') = The\ value\ at\ the\ previous\ state.$
  - In the above equation, we are taking the max of the complete values because the agent tries to find the optimal solution always.

The Bellman Equation

- So now, using the Bellman equation, we will find value at each state of the given environment.

- We will start from the block, which is next to the target block.

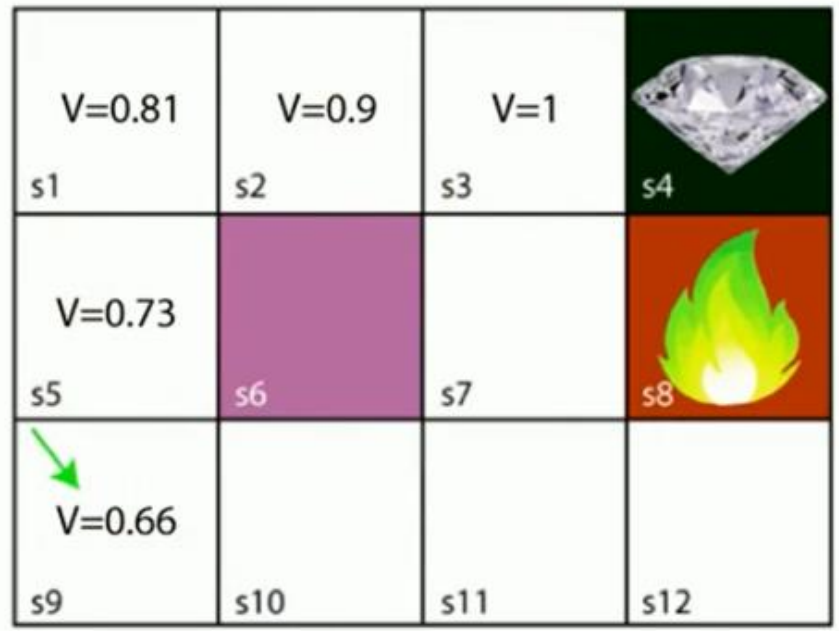
- **For S3 block:**
- $V(s3) = \max[R(s,a) + \gamma V(s`)]$,
- here $V(s') = 0$ because there is no further state to move.
- $V(s3) = \max[R(s,a)] => V(s3) = \max[1] => V(s3) = 1.$



VIT-AP UNIVERSITY

The Bellman Equation

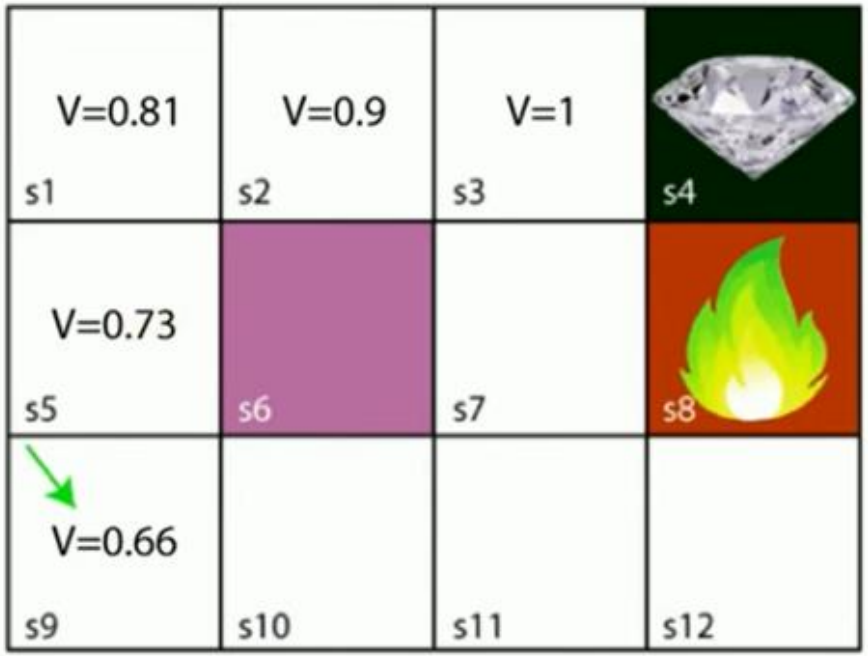
- **For S2 block:**
- $V(s2) = \max[R(s,a) + \gamma V(s`)]$,
- here $\gamma = 0.9, V(s') = 1, and\ R(s,a) = 0$, because there is no reward at this state.
- $V(s2) = \max[0.9(1)] => V(s2) = \max[0.9] => V(s2) = 0.9$



The Bellman Equation

- **For S1 block:**
- $V(s1) = \max[R(s,a) + \gamma V(s`)]$,
- here $\gamma = 0.9(lets), V(s') = 0.9, and\ R(s,a) = 0$, because there is no reward at this state also.
- $V(s1) = \max[0.9(0.9)] => V(s1) = \max[0.81] =>$ $V(s1) = 0.81$

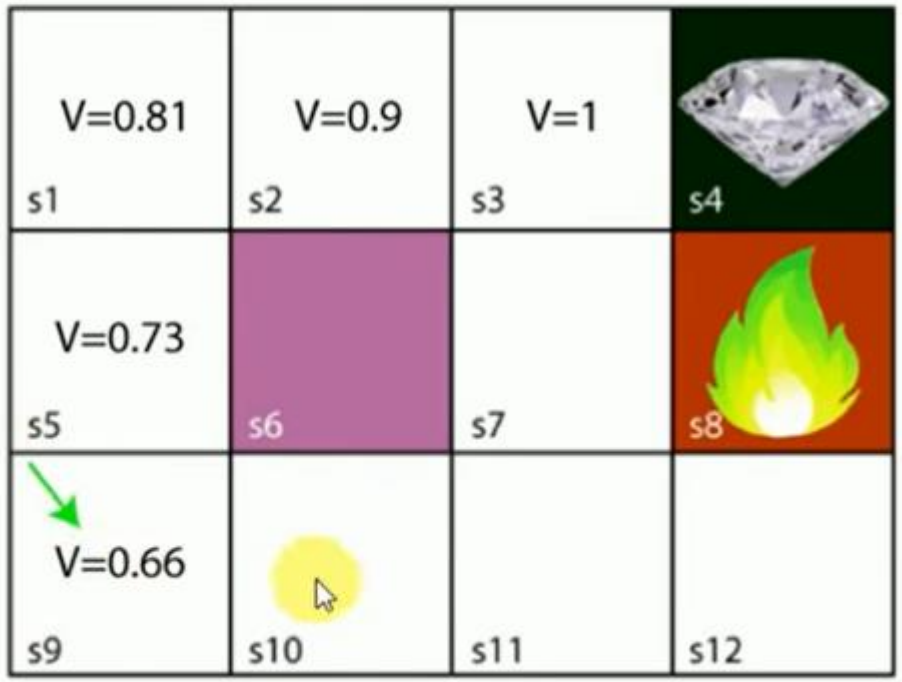
- **For S5 block:**
- $V(s5) = \max[R(s,a) + \gamma V(s`)]$,
- here $\gamma = 0.9(lets), V(s') = 0.81, and\ R(s,a) = 0$, because there is no reward at this state also.
- $V(s5) = \max[0.9(0.81)] => V(s5) = \max[0.73] =>$ $V(s5) = 0.73$



The Bellman Equation

- **For S9 block:**

- $V(s9) = \max[R(s,a) + \gamma V(s`)]$,
- here $\gamma = 0.9(lets), V(s') = 0.73, and\ R(s,a) = 0$, because there is no reward at this state also.
- $V(s9) = \max[0.9(0.73)] => V(s9) = \max[0.66] => V(s4) = 0.66$

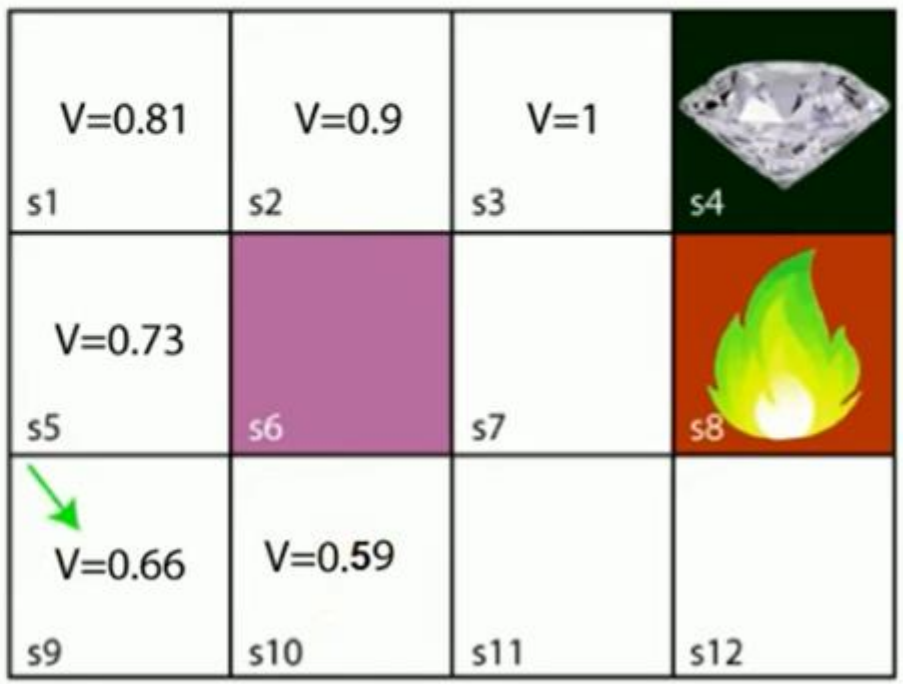


VIT-AP UNIVERSITY

The Bellman Equation

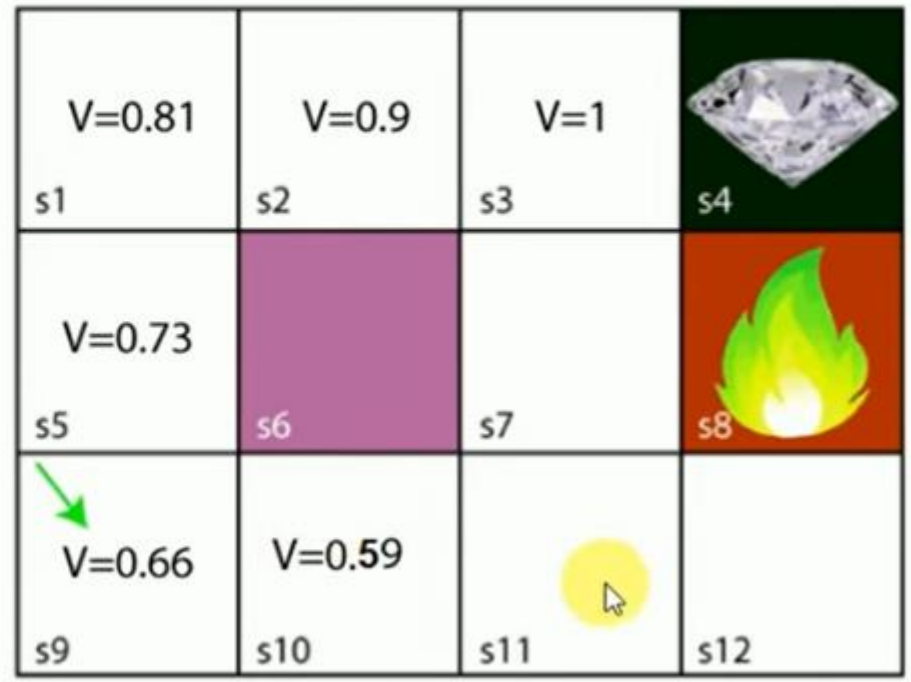
- **For S10 block:**

- $V(s10) = \max[R(s,a) + \gamma V(s`)]$,
- here $\gamma = 0.9, V(s') = 0.66$ *and* $R(s,a) = 0$, because there is no reward at this state also.
- $V(s10) = \max[0.9(0.66)] => V(s10) = \max[0.59] =>$ $V(s10) = 0.59$



- **For S11 block:**

- $V(s11) = \max[R(s,a) + \gamma V(s`)]$,
- here $\gamma = 0.9, V(s') = 0.59$ *and* $R(s,a) = 0$, because there is no reward at this state also.
- $V(s11) = \max[0.9(0.59)] \Rightarrow V(s11) = \max[0.53] \Rightarrow V(s11) = 0.53$

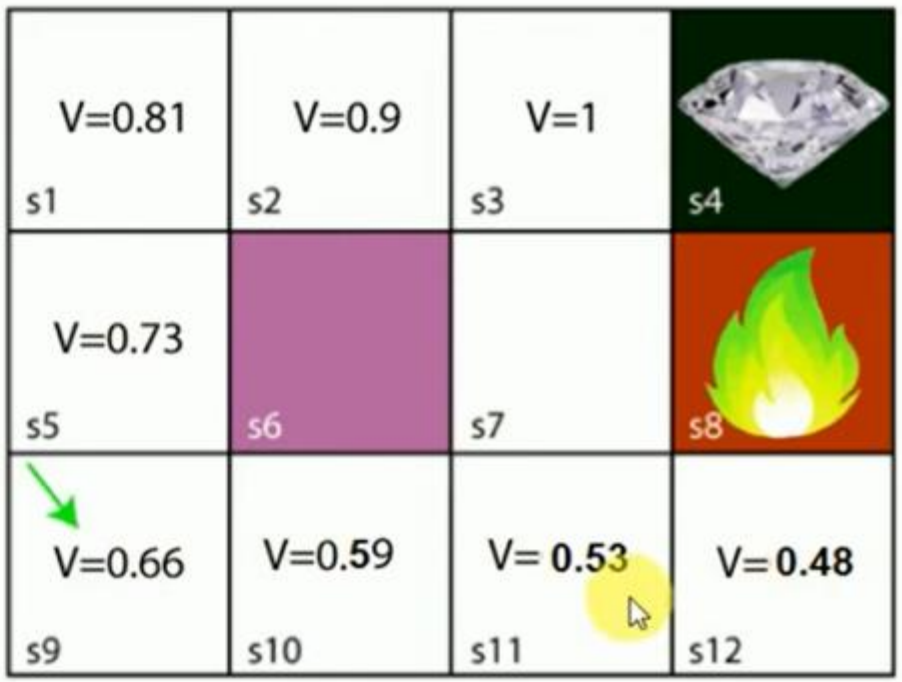


VIT-AP UNIVERSITY

The Bellman Equation

- **For S12 block:**

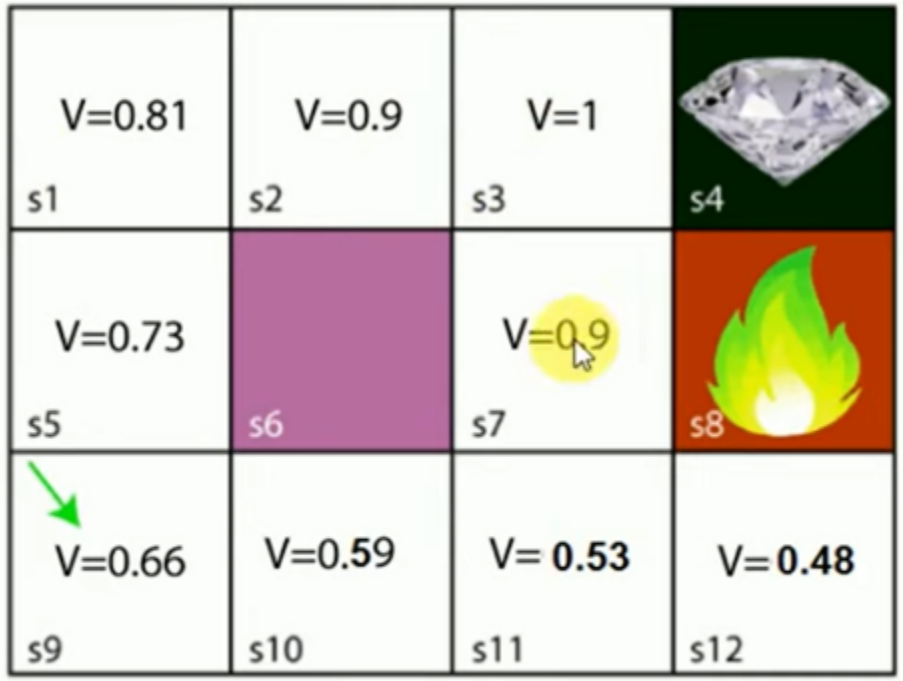
- $V(s12) = \max[R(s,a) + \gamma V(s`)]$,
- here $\gamma = 0.9, V(s') = 0.59$ and $R(s,a) = 0$, because there is no reward at this state also.
- $V(s12) = \max[0.9(0.53)] => V(s12) = \max[0.48] => V(s12) = 0.48$



The Bellman Equation

- **For S7 block:**

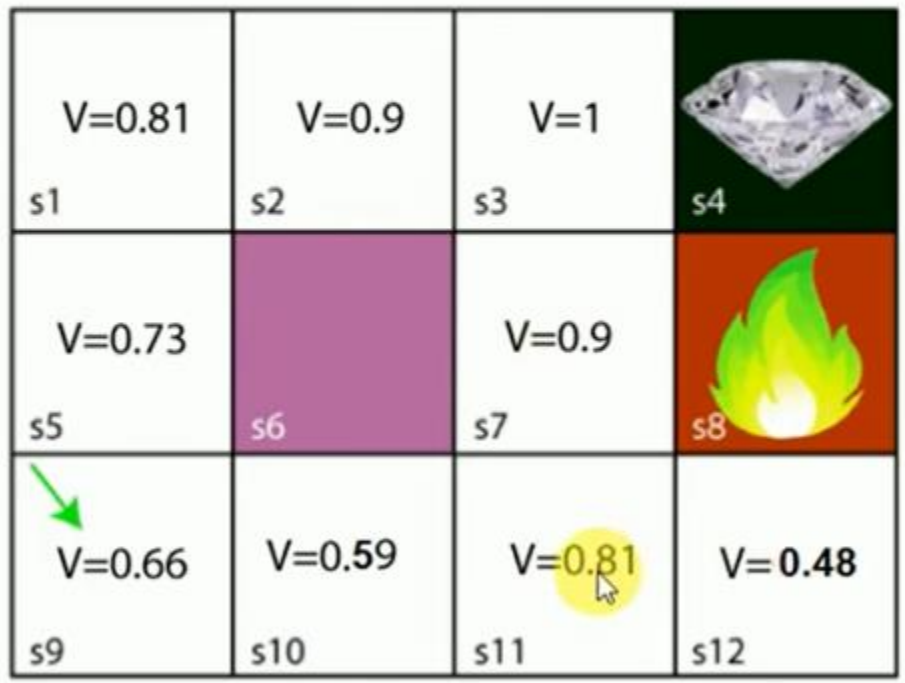
- $V(s7) = \max[R(s,a) + \gamma V(s`)]$,
- here $\gamma = 0.9, V(s') = 1\ and\ R(s,a) = 0$, because there is no reward at this state also.
- $V(s7) = \max[0.9(1)] => V(s7) = \max[0.9] => V(s7) = 0.9$



The Bellman Equation

- **For S11 block:**

- $V(s11) = \max[R(s,a) + \gamma V(s`)]$,
- here $\gamma = 0.9, V(s') = 0.9$ *and* $R(s,a) = 0$, because there is no reward at this state also.
- $V(s11) = \max[0.9(0.9)] => V(s11) = \max[0.81] = > V(s11) = 0.81$

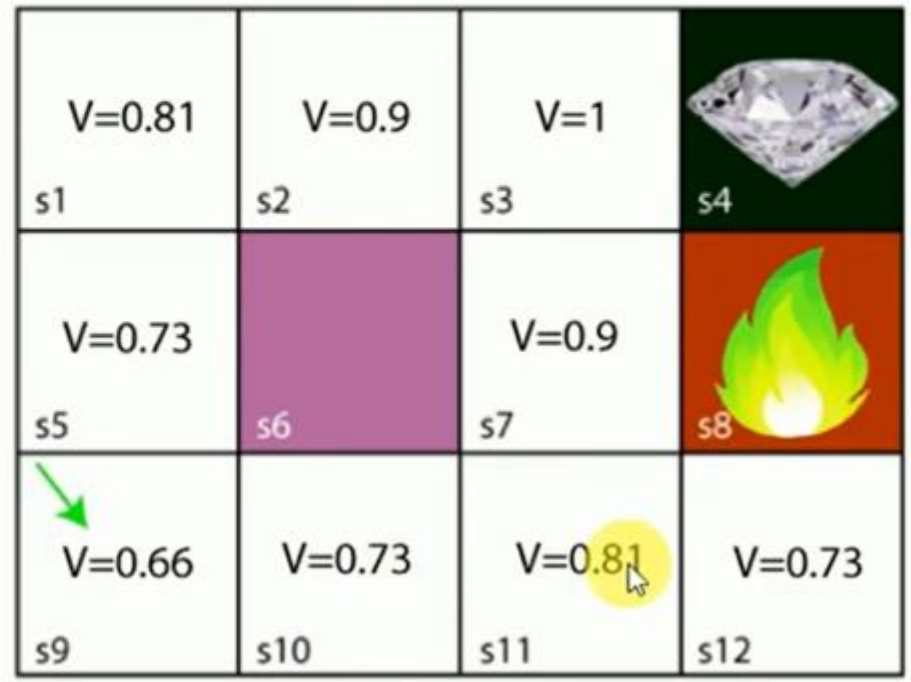


VIT-AP UNIVERSITY

The Bellman Equation

- **For S12 block:**

- $V(s12) = \max[R(s,a) + \gamma V(s`)]$,
- here $\gamma = 0.9, V(s') = 0.81$ *and* $R(s,a) = 0$, because there is no reward at this state also.
- $V(s12) = \max[0.9(0.81)] => V(s12) = \max[0.73] = > V(s12) = 0.73$



VIT-AP UNIVERSITY

Thank You