Thursday Learning Hour

Reinforcement Learning

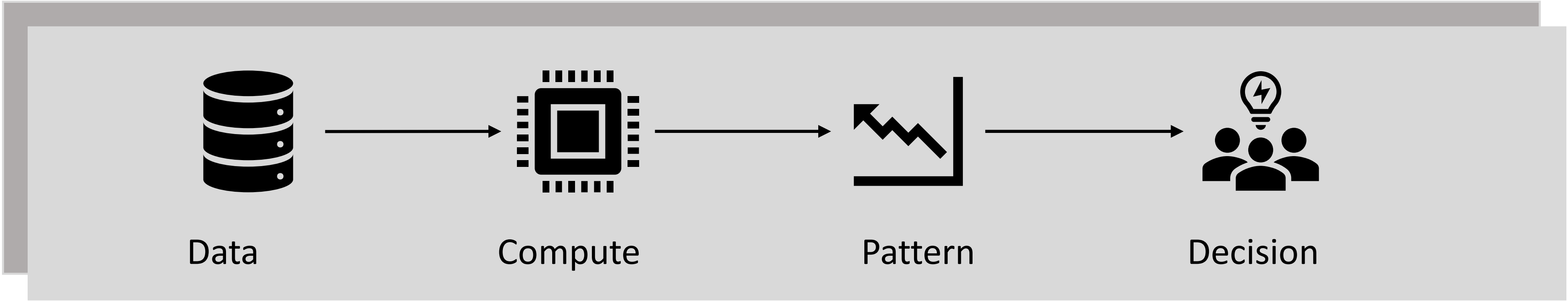
By

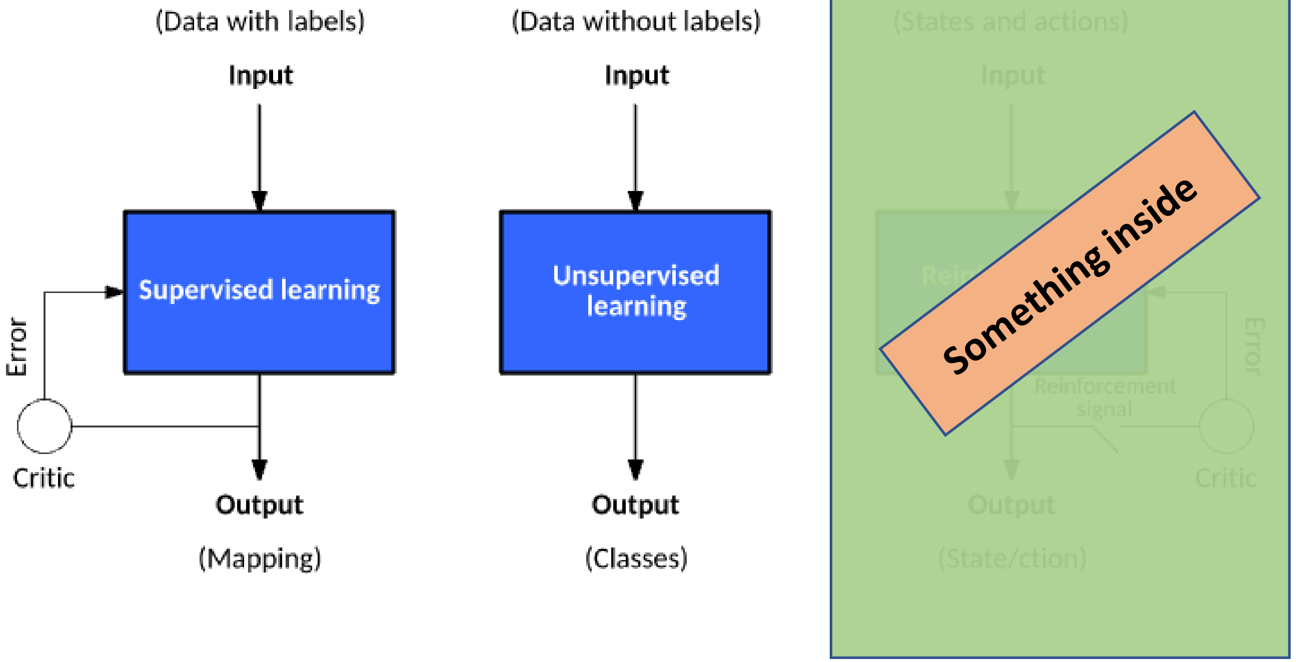
Prabakaran Chandran

Agenda:

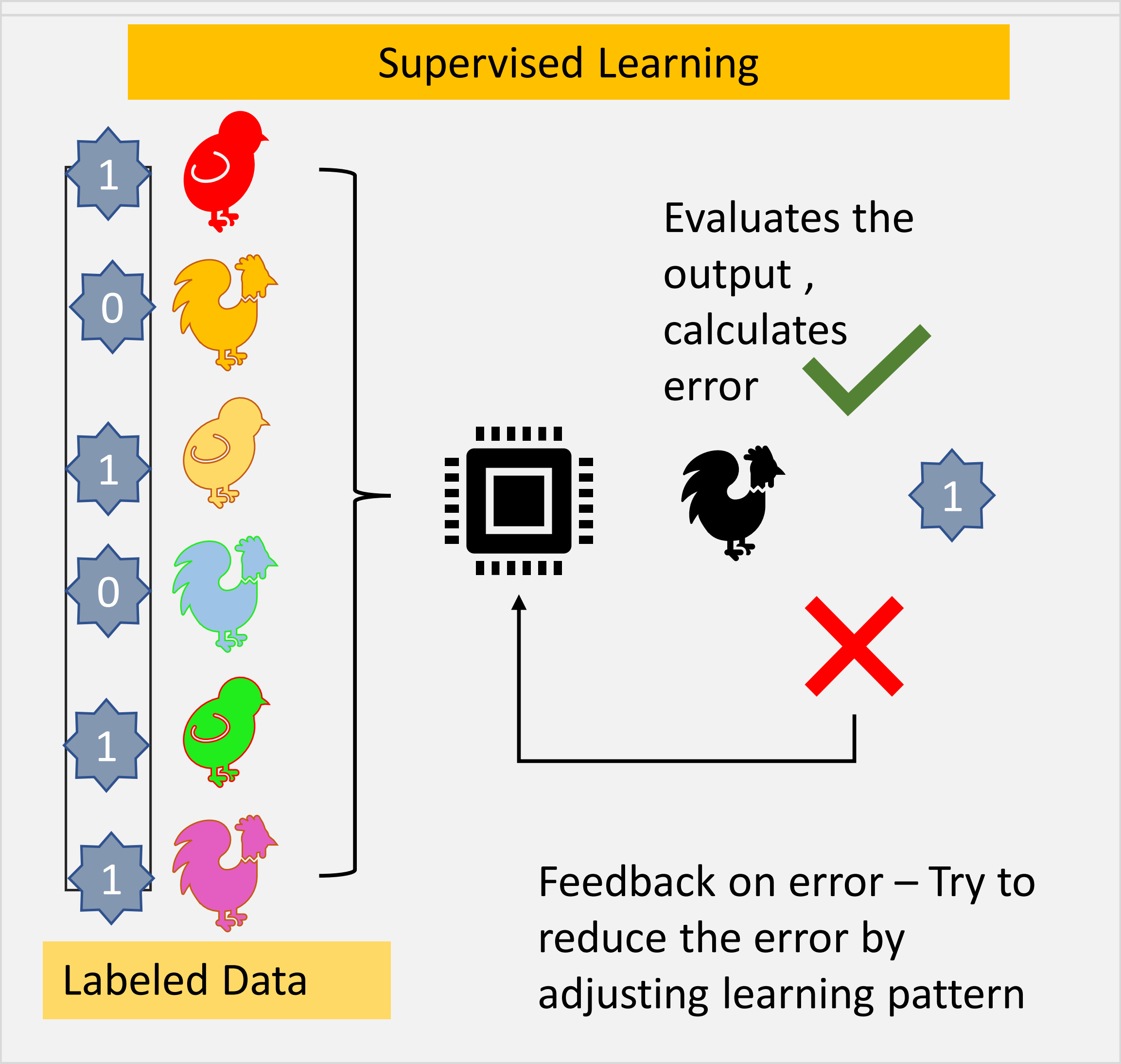
- An overview on Machine learning paradigm
- Conventional Machine learning vs Reinforcement learning
- Key Concepts of Reinforcement learning
- Foundation of RL – Markov Family
- Components of RL
- Applications across domains

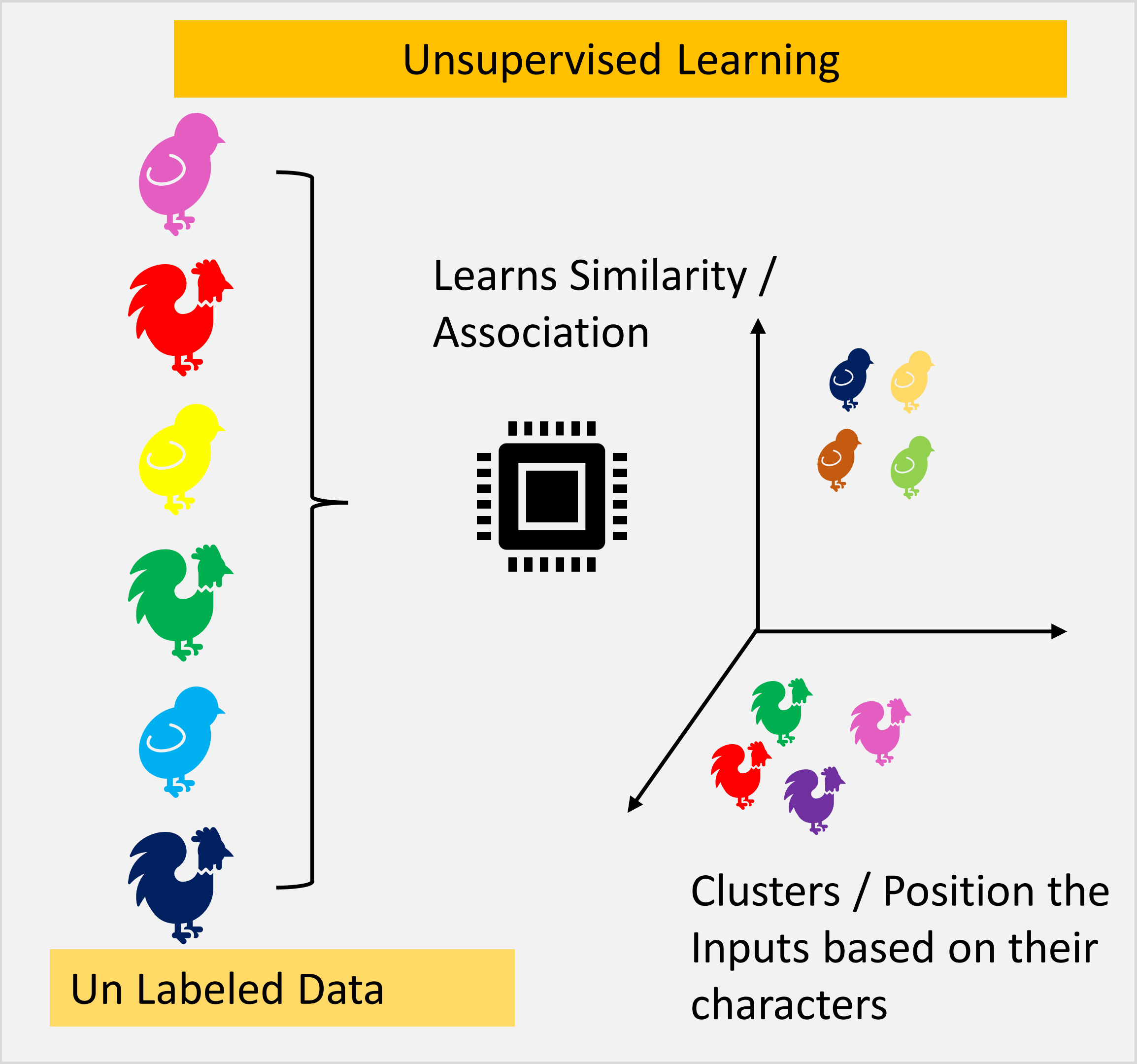
Machine learning: Conventional ML vs Reinforcement Learning



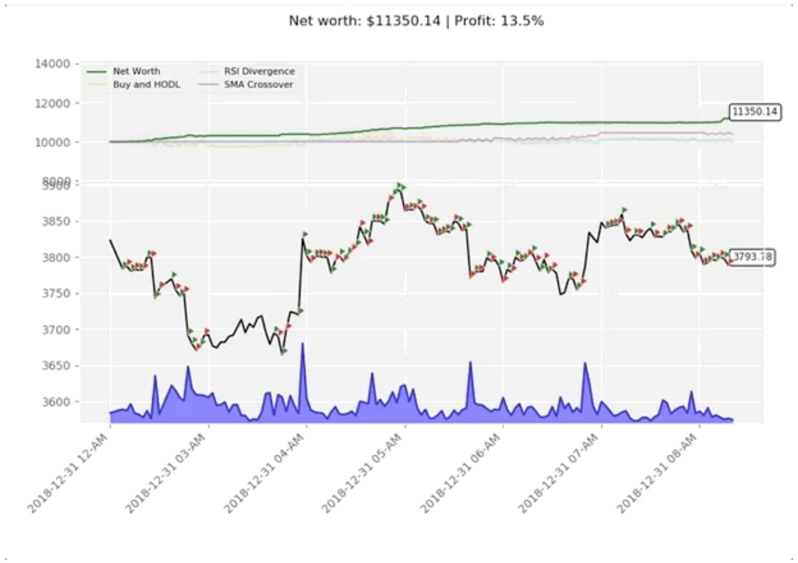


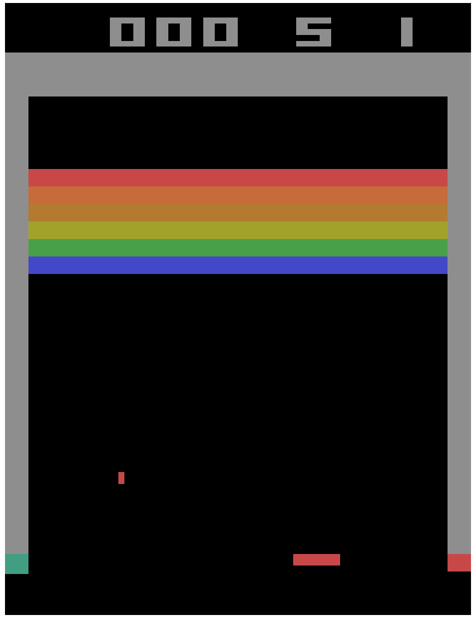
Machine learning : Supervised and Unsupervised

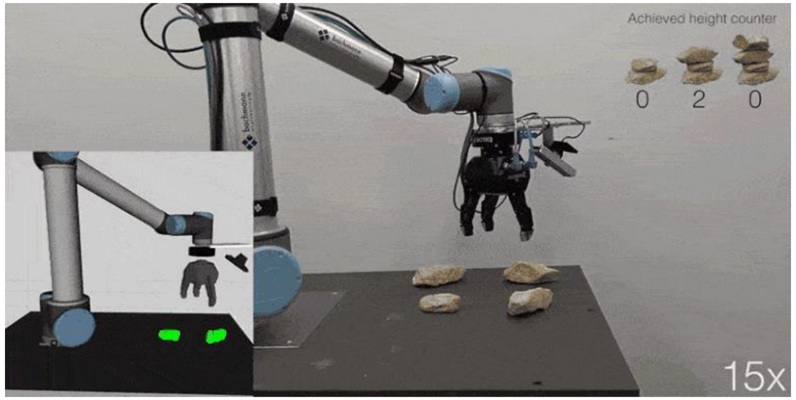




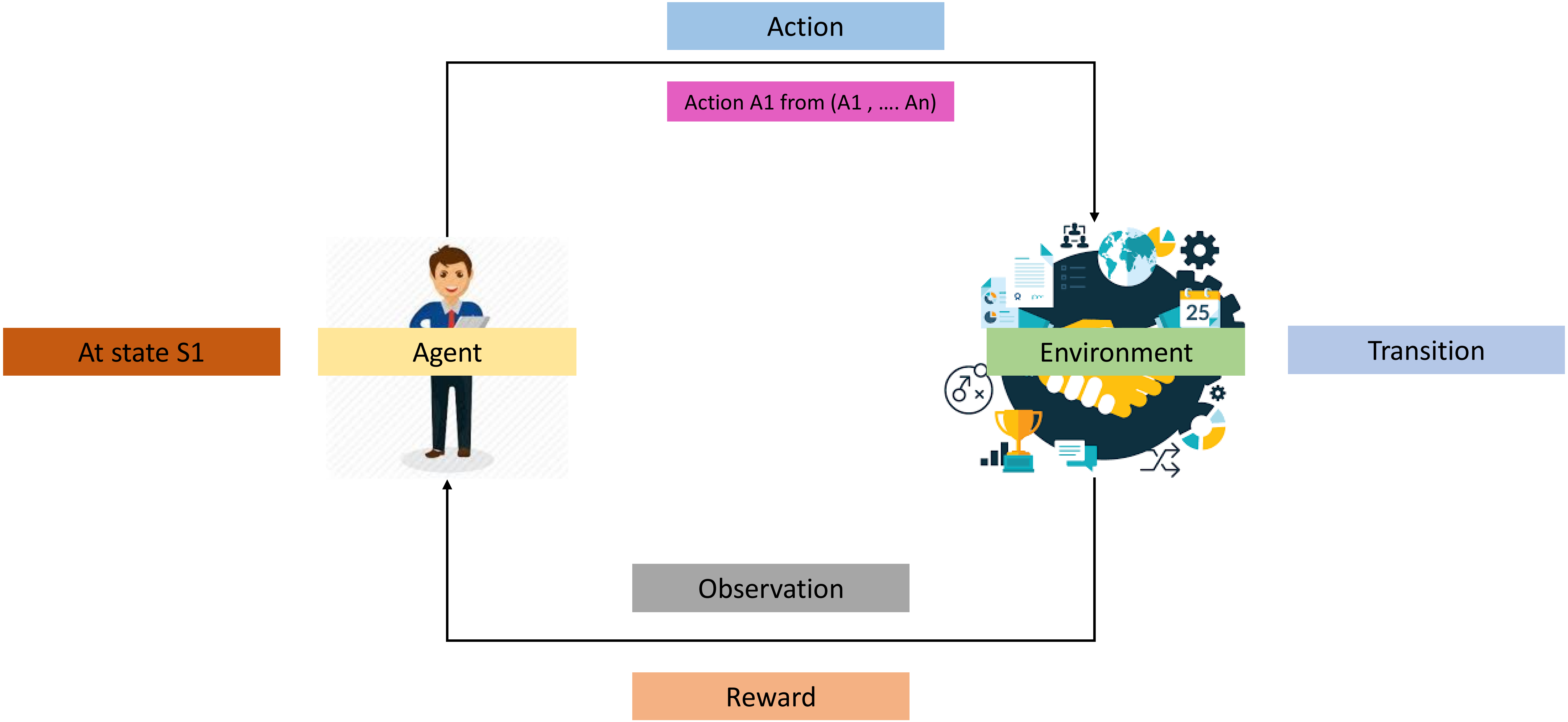
What happens in these Scenarios ?:







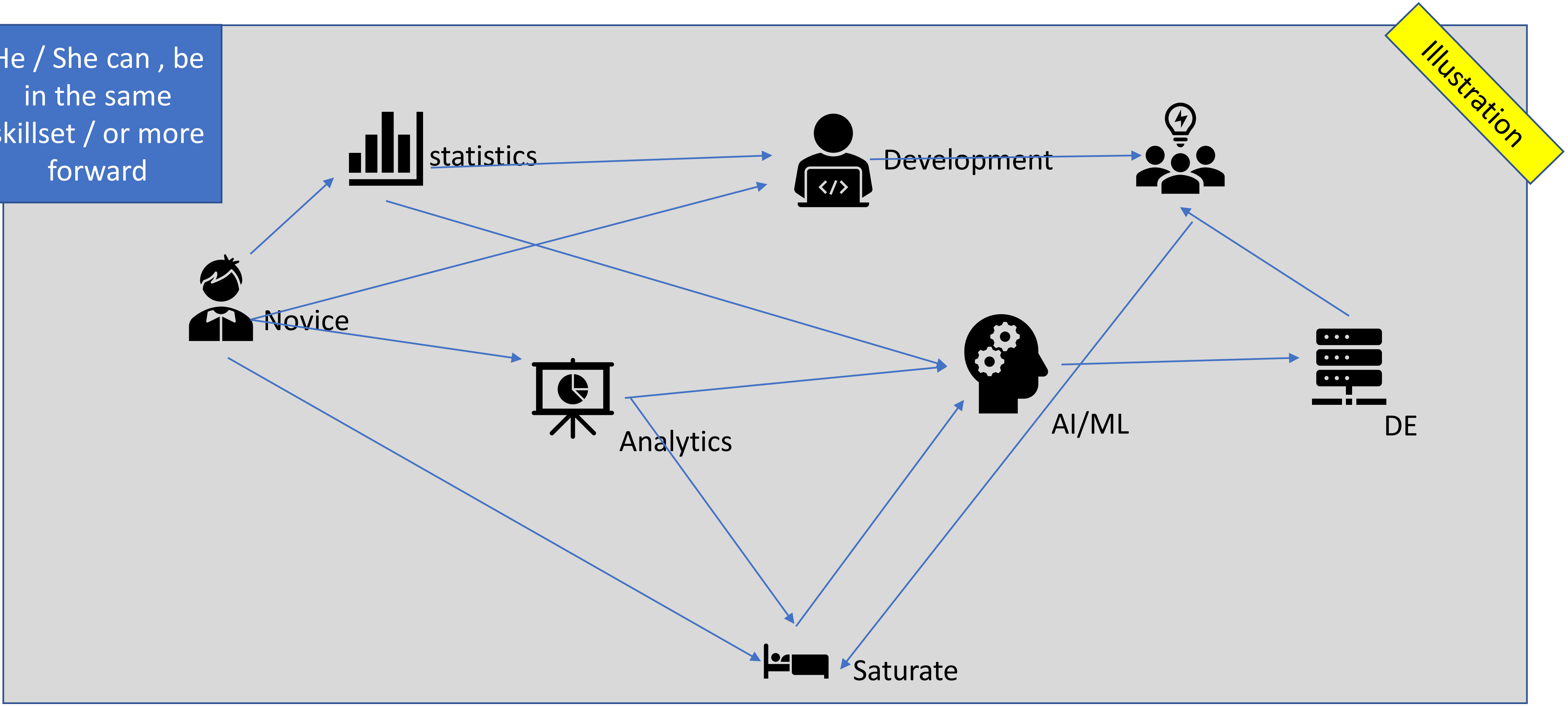
Reinforcement learning



What is Reinforcement Learning?

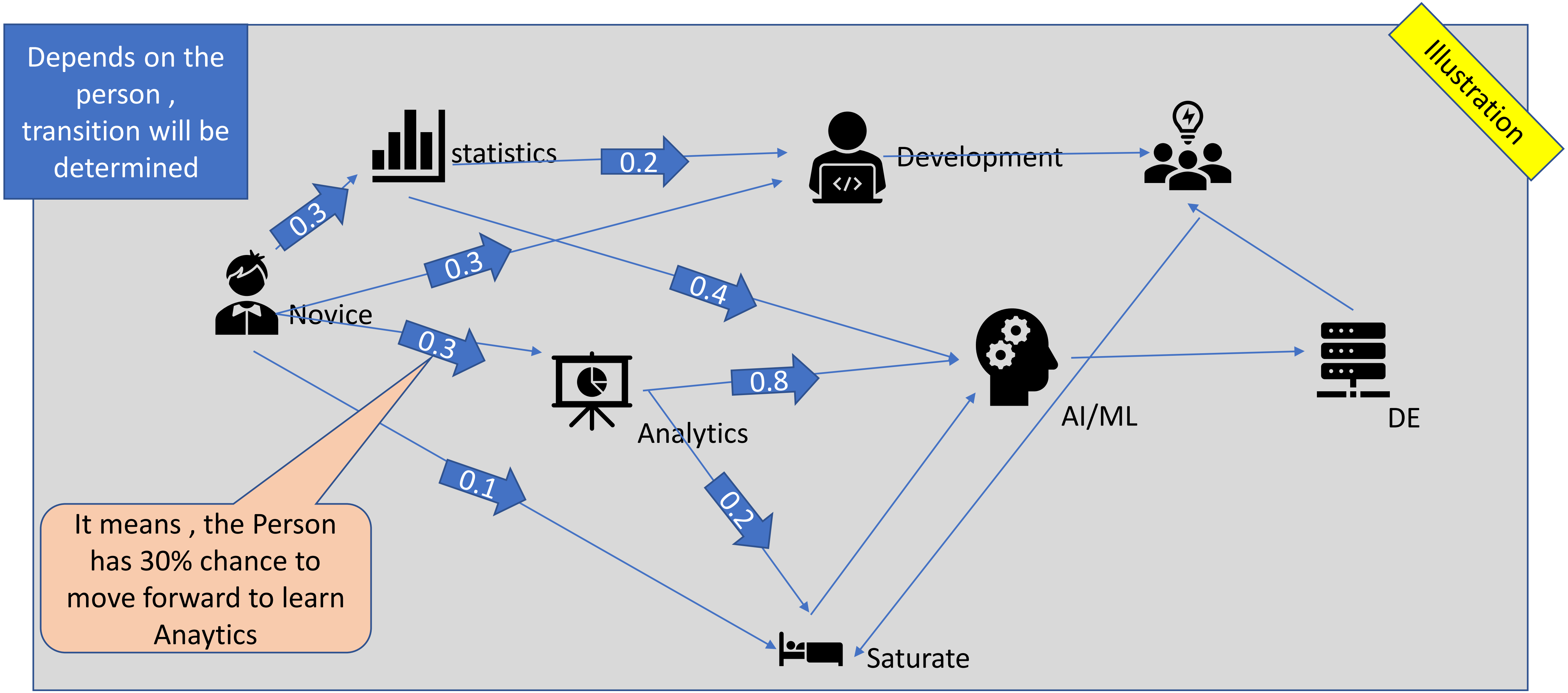
Consider a Person and his Journey in Data & AI Domain

He / She can , be in the same skillset / or more forward



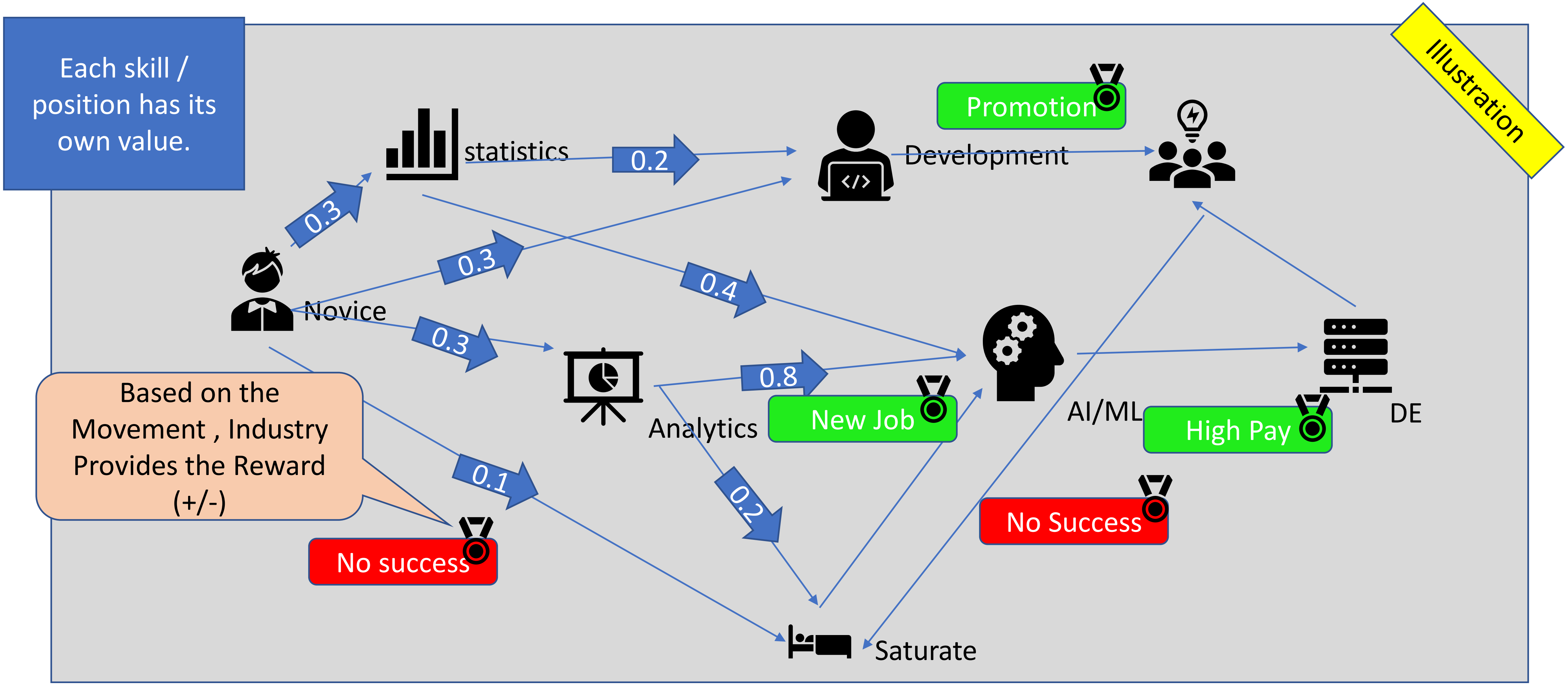
What is Reinforcement Learning?

Person has a set of skills to learn – Based on his Behavior the person selects the skill or saturation



What is Reinforcement Learning?

Based on the skill / position the person stays , Industry gives him/her a Reward



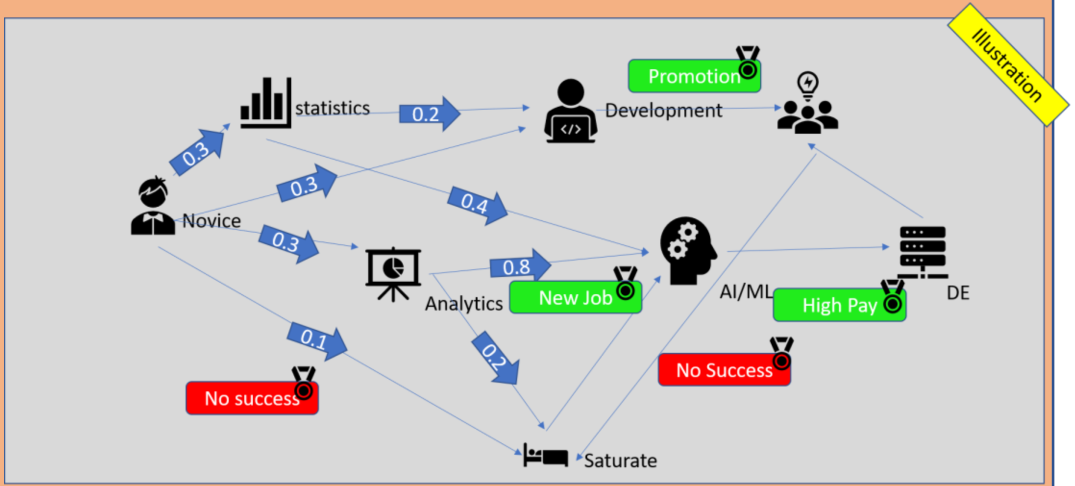
What is Reinforcement Learning?

To become successful in the industry , the person change his behavior - By learning and understanding

Here , I denote

- Person → Agent
- Each skill / position →State
- Data & AI industry → Environment
- Probability → Transition Probability
- Promotion , Failure → Reward
- Scope of Each position →value of the state
- Skill / Position transition behavior of the person →Policy

- What we need is optimal behavior / optimal policy to have more success in the environment

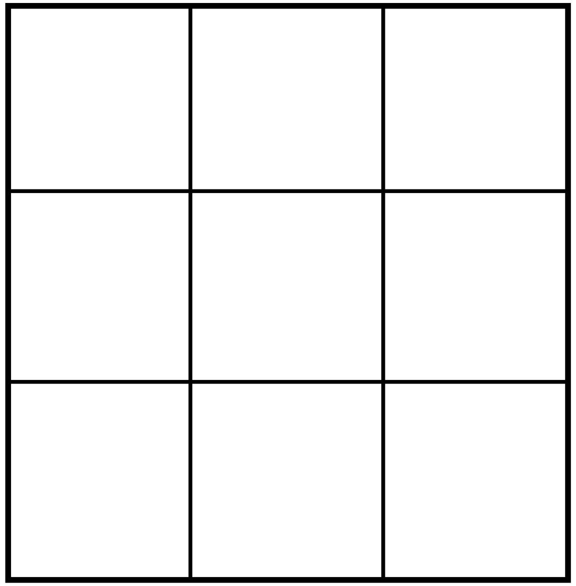


- How did I structure this process to capture these interactions ?
    - Markov Family – Helped me (Lets dive into small math part)

- By selecting correct decisions / actions in each state , the person can build his optimal policy (optimal behavior) which can give him a great success

- This Selection cannot be achieved directly , its by error and trail (Learning) ---→ Reinforcement learning

Markov Family of Processes builds the RL environment



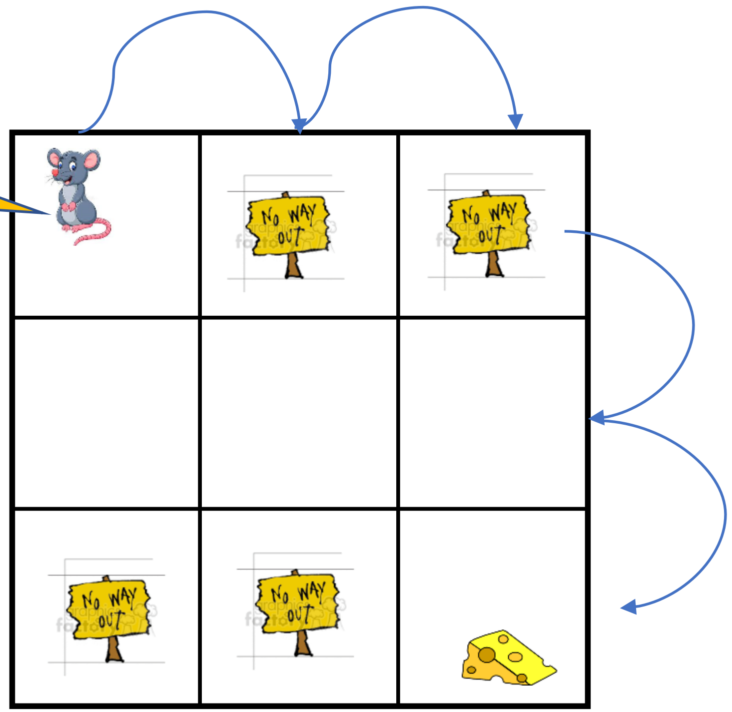
Space

Meet master. JD , He don't know how to get the cheese , need to learn

Posses Markov Property

$$P_{(s_{t+1}|s_t)} = P_{(s_{t+1}|s_t,....s1)}$$

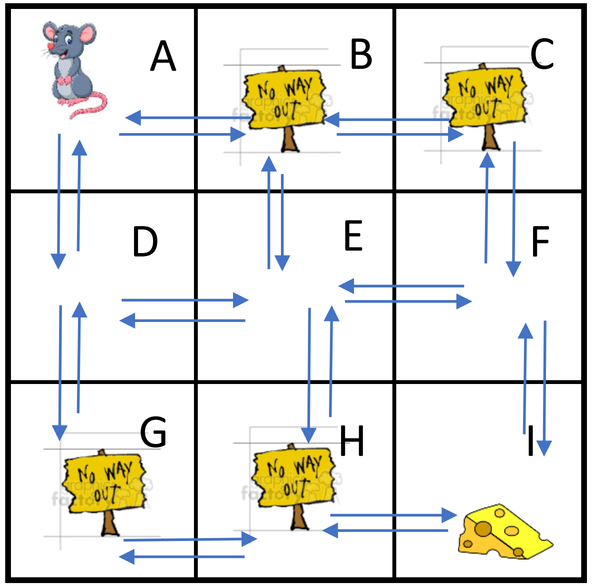
Markov is nothing but Future state is depending on current state , not the History



State Space

Markov Property makes each state as memoryless

Markov Property → Markov Process → Markov Reward Process → Markov Decision Process:



Possible State change (process / Chain)

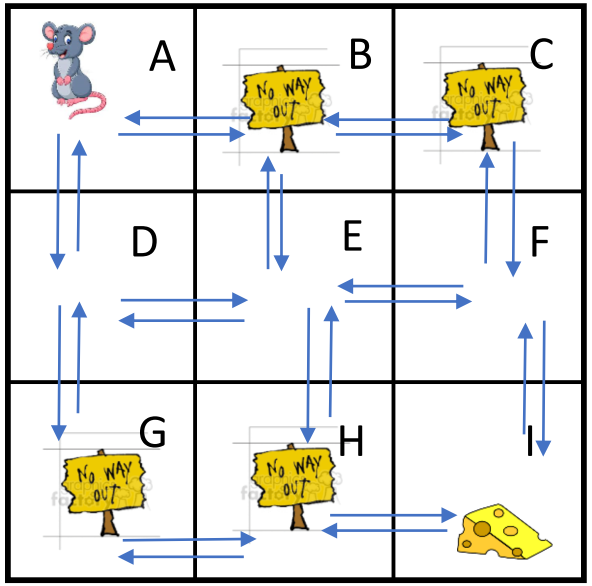
Transition Matrix

Markov Process

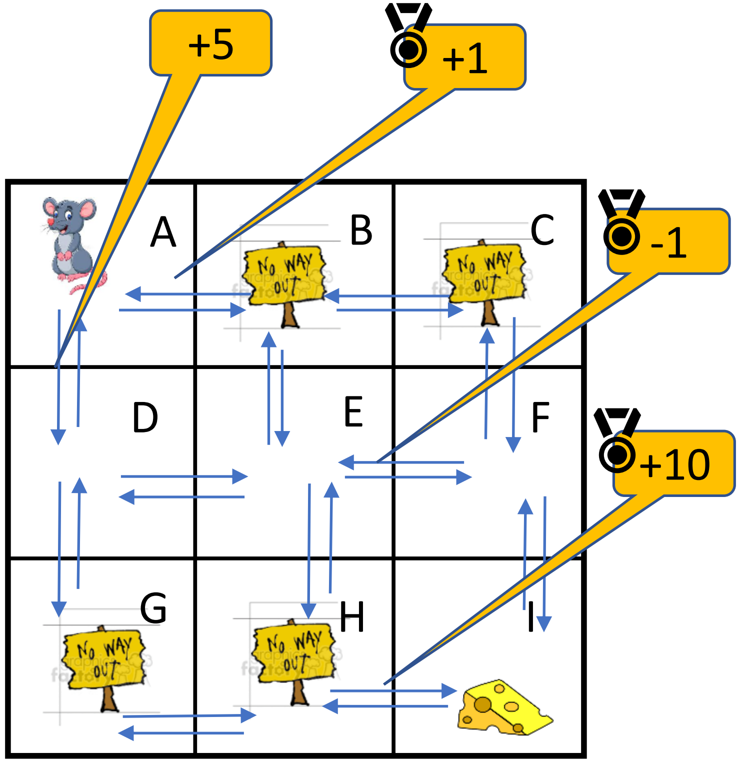
	A	B	C	D	E	F	G	H	I
A	0.2	0.1		0.7					
B	0.5	0.4	0.1						
C		0.4	0.2			0.4			
D	0.1			0.2	0.6		0.1		
E		0.1		0.3	0.2	0.3		0.1	
F			0.1		0.1	0.1			0.6
G					0.6		0.2	0.2	
H					0.1		0.1	0.1	0.7
I									

Out of 81 transition probabilities , only 29 are Possible in this state space.

Markov Reward Process:



	A	B	C	D	E	F	G	H	I
A	0.2	0.1		0.7					
B	0.5	0.4	0.1						
C		0.4	0.2			0.4			
D	0.1			0.2	0.6		0.1		
E		0.1		0.3	0.2	0.3		0.1	
F			0.1		0.1	0.1			0.6
G					0.6		0.2	0.2	
H					0.1		0.1	0.1	0.7
I									

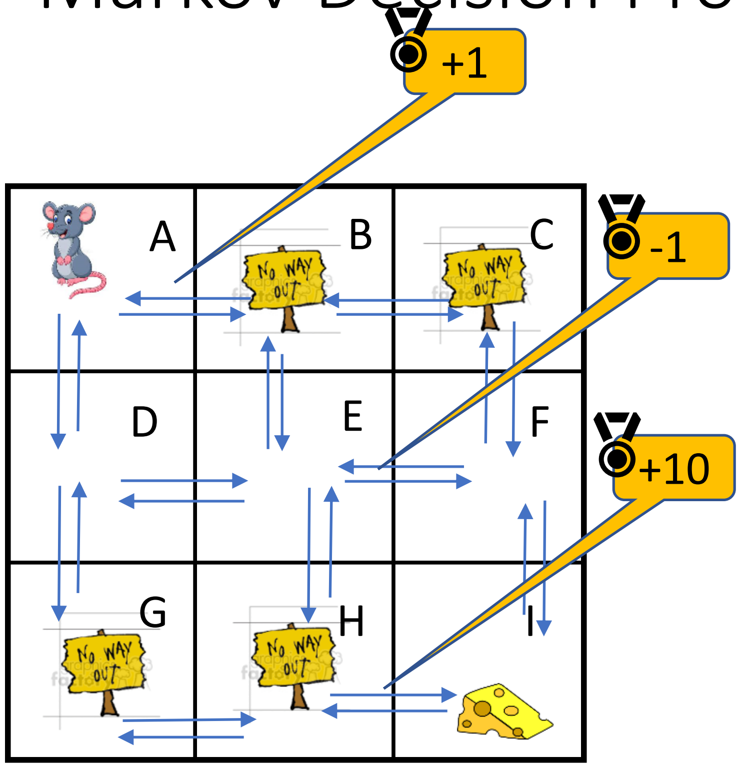


Possible State change (process / Chain)

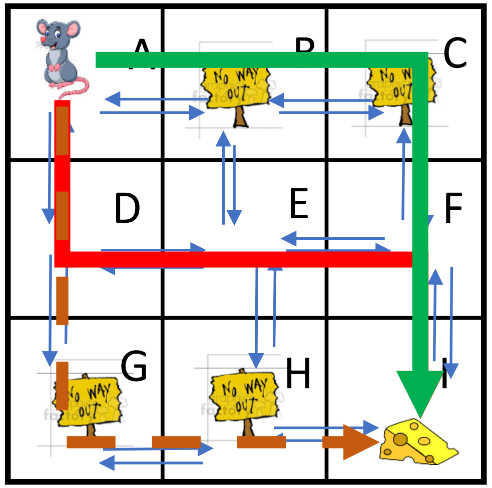
Each Transition will have its own reward point , Reward point can be positive or negative depends on their properties.
Each State can have a value , which is expected return by being in the state.
Expected Return = Reward at the current state + d(Expected Return of Previous state)

Discount factor – to avoid infinity and significant to Current State.

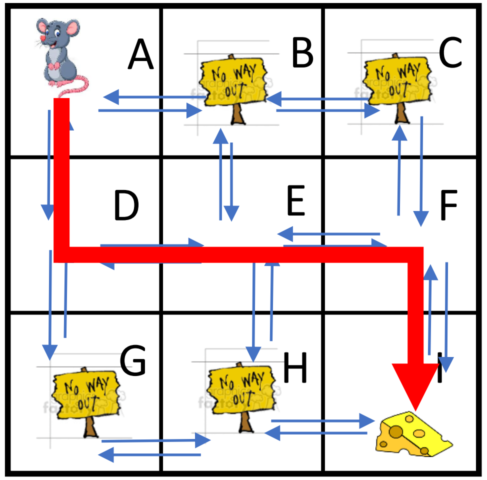
Markov Decision Process:



- At A (B , Right) and A(D, down) are the possible actions
- JD can select any one of the action (Decision he has to take)



Possible Decisions by selecting the Actions at each state.



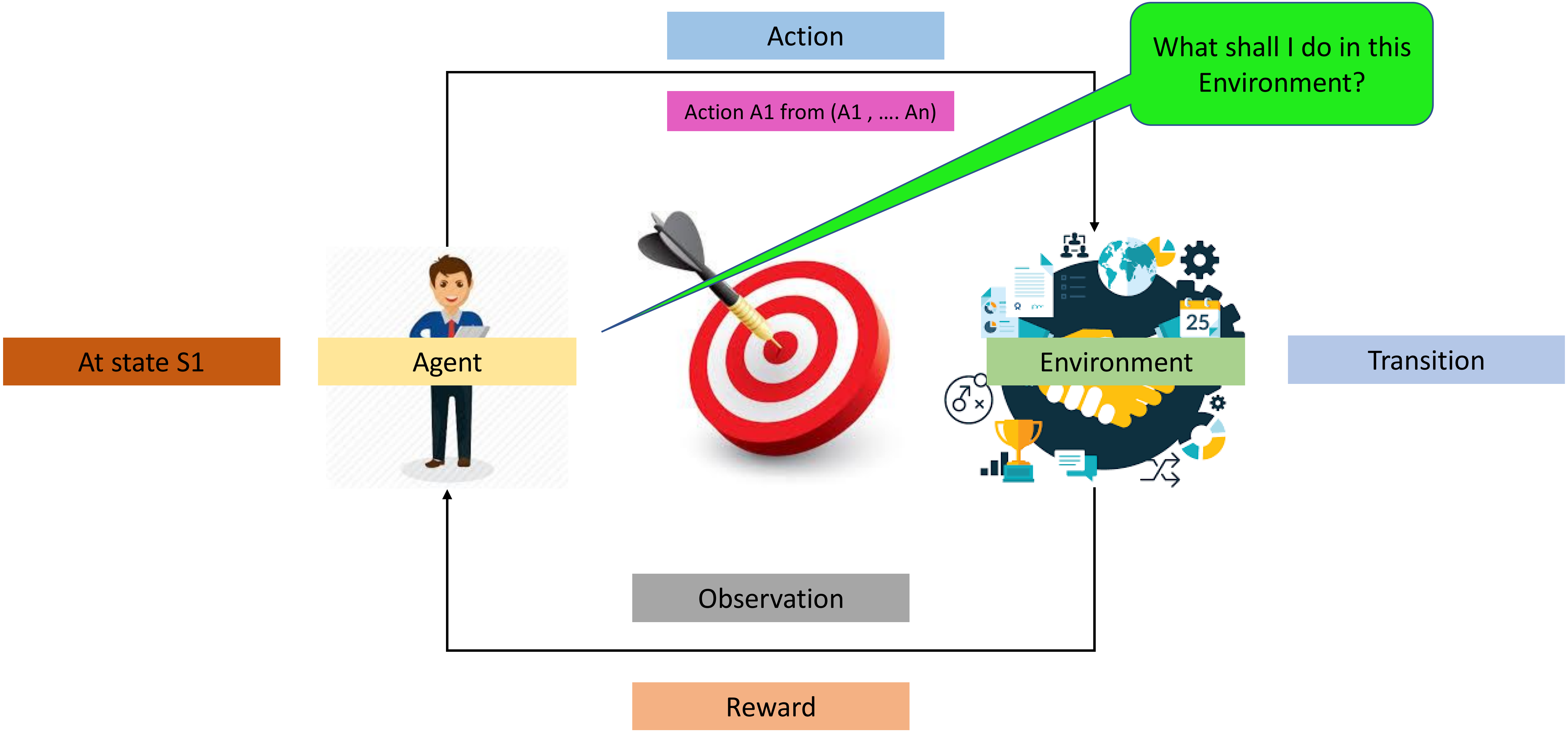
This is what we Jeffery need to follow to have more reward.

How can he learn this ??

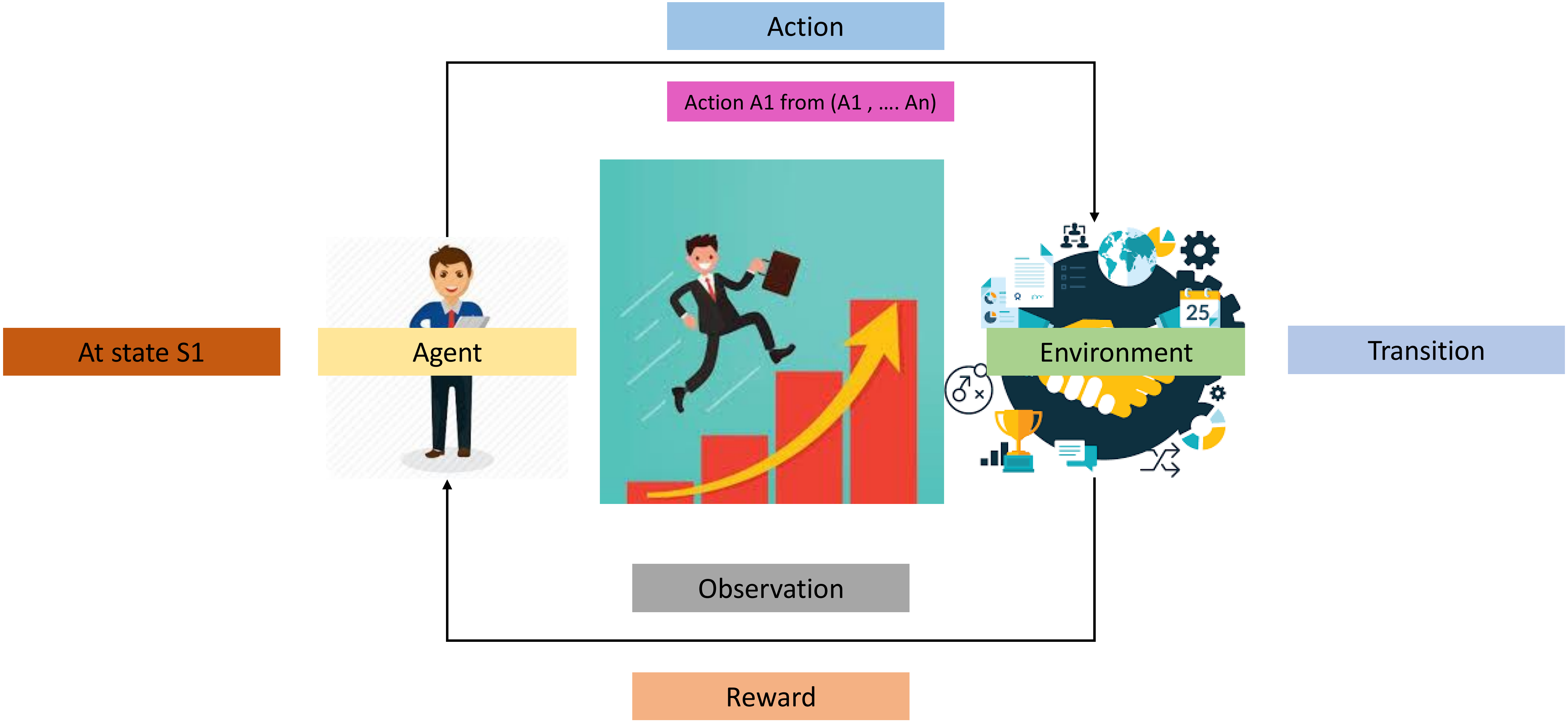
By solving this MDP Enviornment

He can take any Decision from the available action space , but there can be a negative reward too. This selection of action depends on Jeffery's Behavior (Policy)

Reinforcement learning.



Reinforcement learning.



Reinforcement learning.

I need to increase my Rewards

I need a Strategy

Action

Action A1 from (A1 , …. An)

At state S1

Agent

Environment

Transition

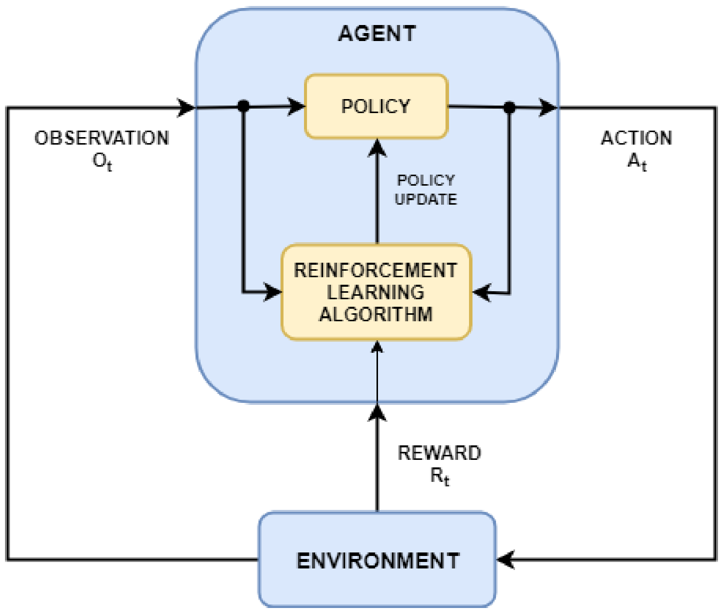
Observation

Reward

To reach the optimal policy : Strategy Making

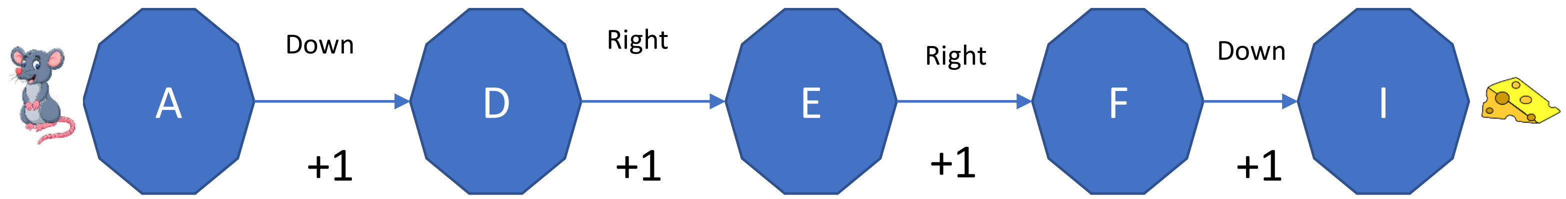
Let's Define Policy :

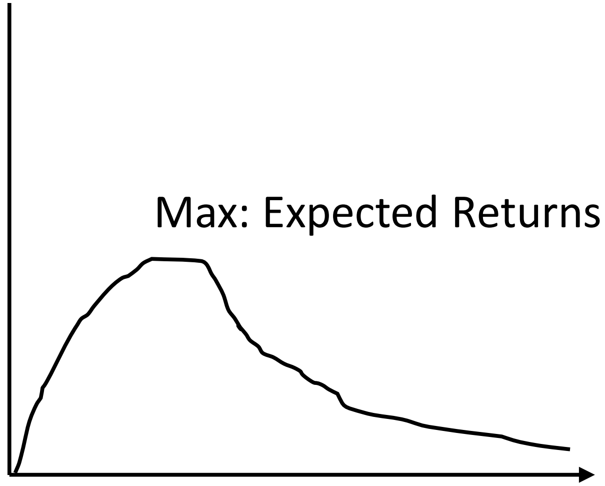
- π; $s \rightarrow Pr(A/s)$, where $s \in S$
- *In simple word for each s → a ; A → down , B → left , D → right*
- π is the policy here.
- It's a mapping from states to the (probabilistically) best
- actions for those states.



Optimal policy : $\pi* = \text{arg}max\ E(R/\pi)$, the policy which gives more return

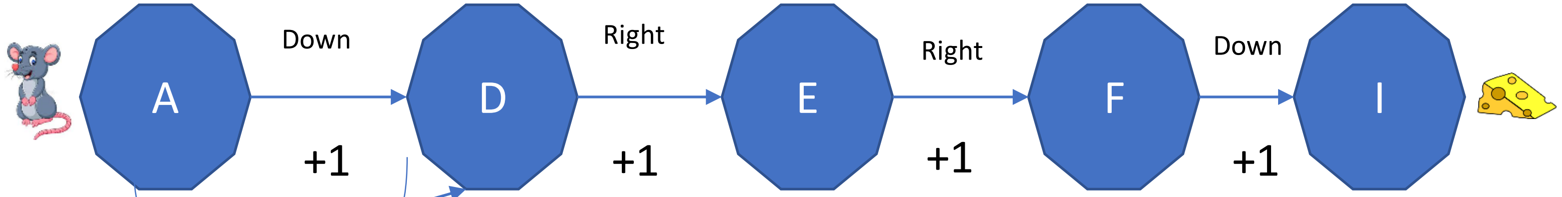
1. How Can we evaluate the Policies ? And select the Optimal one ?





Policies ------ >

Two Major Functions are used to Evaluate the Policies :



How Good is that to be in D ?
Is this particular state safe to be in ?

Value Function Defines that

V = Expected Return = sum of all rewards
(for now, discount factor is 1)

How Good is that to take a action (down) be in A ?

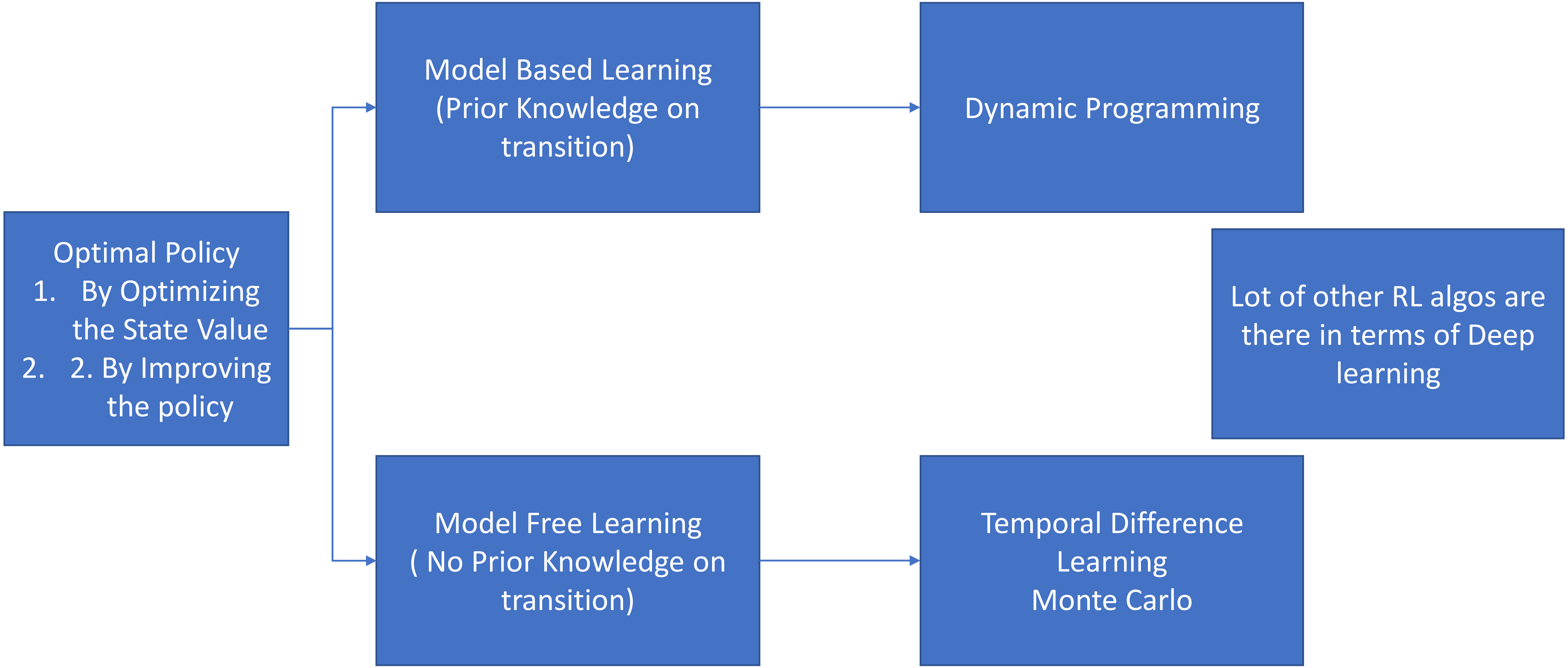
Will this Action and State pair give us more reward?

Q Function Defines that
V = Expected Return = sum of all rewards
(for now, discount factor is 1)

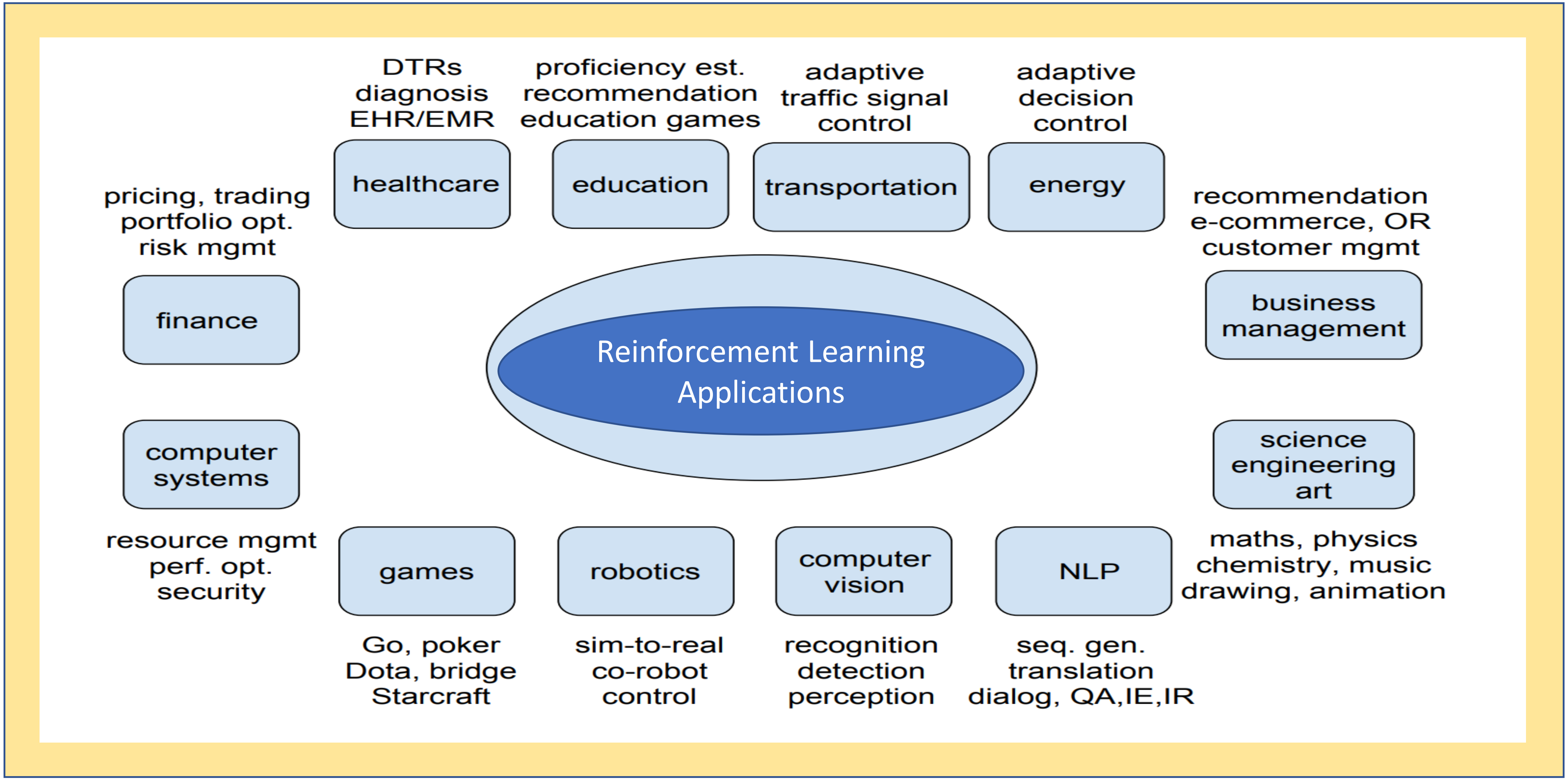
$$V^{\pi}(s) = E_{\pi}\{R_t | s_t = s\} = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, s_t = s\right\},$$

$$Q^{\pi}(s,a) = E_{\pi}\{R_t | s_t = s, a_t = a\} = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, s_t = s, a_t = a\right\}.$$

Road map to RL algorithms: (Classical RLs)



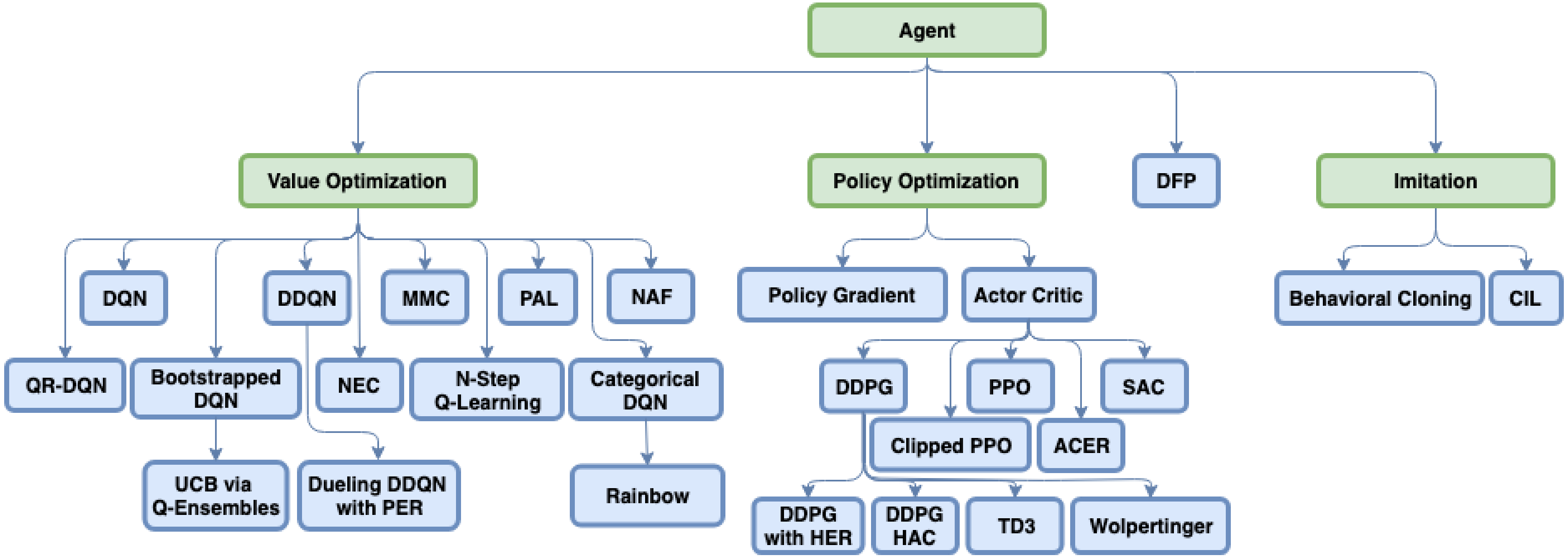
Applications:



Questions:

Appendix:

Deep RL and Other algorithms:



Classical RL process:

Two Functions are used to evaluate the policy

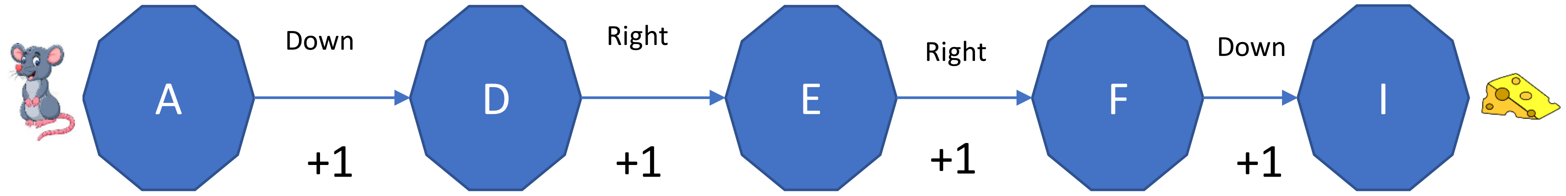
Value Function

Q Function

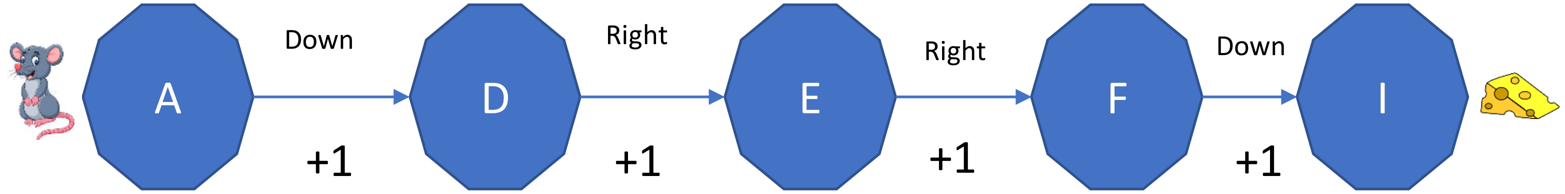
1. **The value function denoted as v(s) under a policy π represents how good a state is for an agent to be in.**

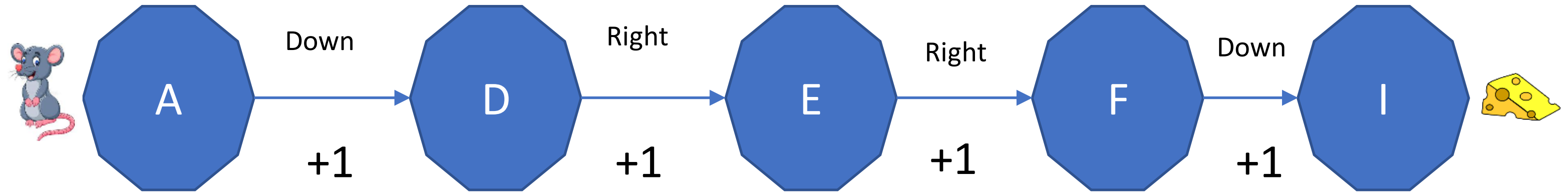
1. Evaluates the State and Action Pair → How good is to take a particular action in a state

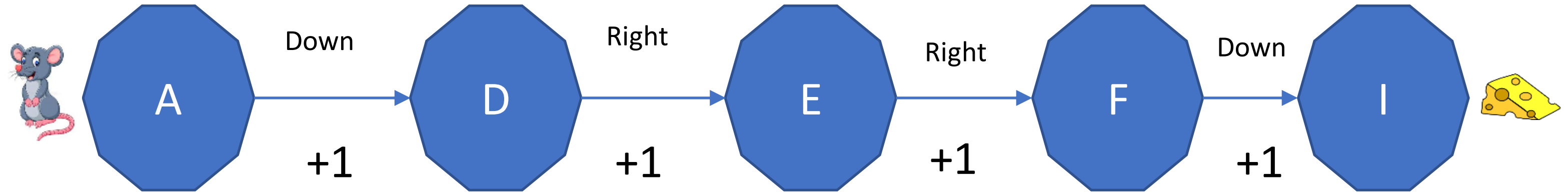
Value Function and Q Function



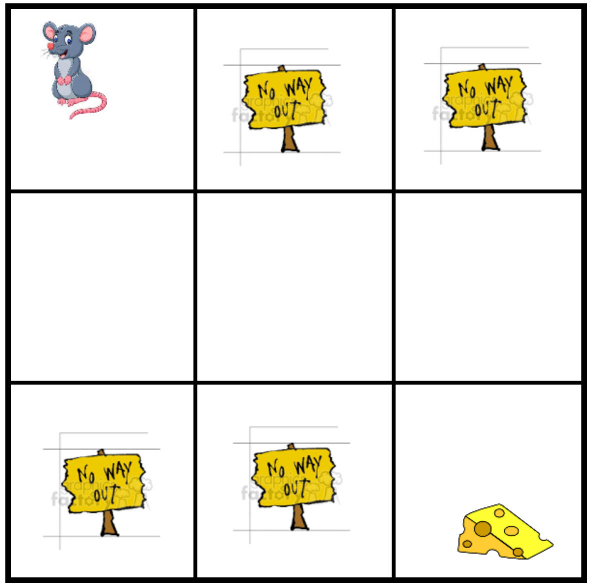
Q function → 4 + 3+ 2

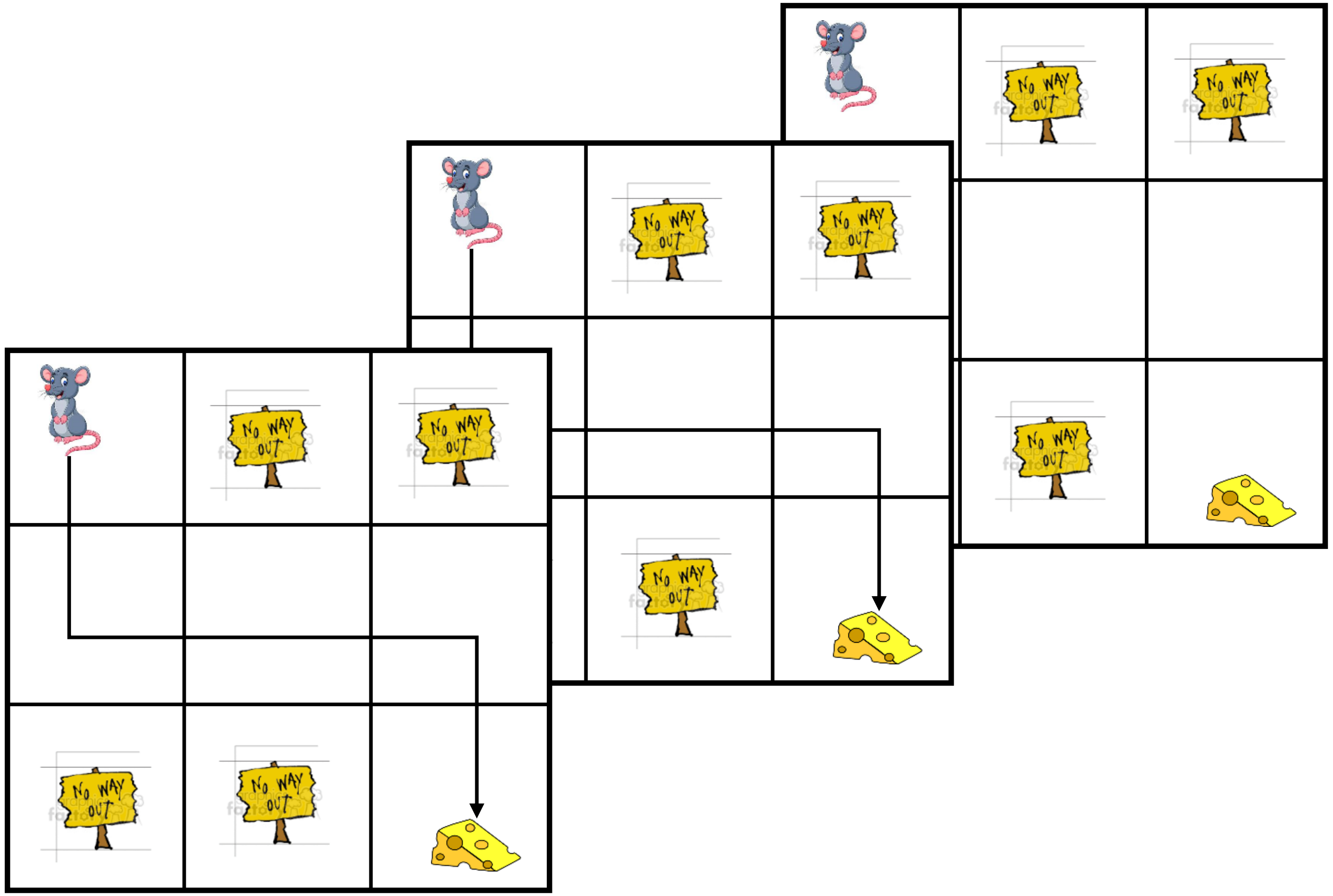






Markov Property → Markov Process → Markov Reward Process → Markov Decision Process:





Reinforcement learning.