# **Upper Confidence Bound**

Definition :- Upper Confidence Bound algorithm is based on the principle of ****optimism in the face of uncertainty****, which is to choose your actions as if the environment (in this case bandit) is as nice as is ****plausibly possible****.

To Understand the Practical meaning of Upper Confidence Bound we need to understand **Multi - Armed Bandit Problem.**

 Imagine, We have a Multiple Armed Bandit Machine. Now, Our objective is to identify which machine can give us a high success rate to earn money.

For this, we need to understand the machine one by one and after a long trail we will understand the profitable machine but for this we need to invest a lot of energy, Time and Money which not a very good idea to practice. So, identify the good machine. We can use Upper Confidence Bound algorithm.

This algorithm take analysis simultaneously with our action and it dynamically find the optimum machine.

If we analyze the flow of our understand the optimum machine then we can say following points,

1. We have d arms.
2. Each time user pulls one arm and which makes one round .
3. At Each Round till n, we are choosing one arm.
4. And at each round till n, we are getting rewards if we are pulling the correct machine for that round. If user pulls the right ARM then, reward increments by 1 or reward remains same( reward increments by 0 ) for that round.
5. Our goal is to maximize the total reward and to identify the optimal Bandit machine.

To achieve above goal we need to follow some processes or set of algorithms,

Step 1 :- At each round till n, we consider two elements for each machine.

Let Consider we have ‘i’ number of machines,

Nj( n ) :- Number of times machine ‘j’ was selected up to n times.

Rj( n ) :- Sum of rewards of machine ‘j’ up to n times.

So, We will make a list,

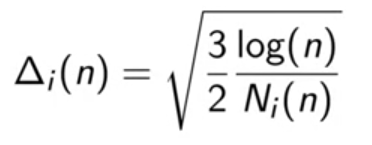
Ni( n ) as no\_of\_selection = [0] \* 10 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

Ri( n ) as reward\_sum = [0] \* 10 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Step 2 : - We compute the average reward for machine ‘i’ up to round n.

Average\_reward = reward\_sum[i] / no\_of\_selection[i]

And we need to compute the Confidence Interval,



Step 3 :- Then, we do UCB = Average\_reward +

Now, we will calculate the UCB for all the machines in each round. But we will select the largest UCB of that round.