# **Thompson Sampling**

Thompson Sampling is same as UCB( Upper Confidence Bound ) the only difference is, In UCB we are calculating the average reward for machine ‘i’ up to round n.

whereas, In Thompson Sampling we are calculating the optimum machine for each round using

**Beta distribution** and then, we are that machine which has highest **Beta distribution** for that round, plus we are adding the reward also for that specific round for all columns.

To Understand the Practical meaning of Thompson Sampling 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 either 1 or no reward then, 0. if we are pulling the correct machine for that round. If user pulls the right ARM then, reward increments by 1 else, we are taking record of each machine if we are not getting any reward..

We are using 2 list,

reward\_list\_1 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0] and reward\_list\_0 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

We have taken 10 -> 0’s for each columns. At every round we will increment each list. Based on condition that reward we got or not.

1. 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 numbers for each machine.

Let Consider we have ‘i’ number of machines,

Ni1( n ) :- Number of times machine ‘i’ was selected up to n times but we got the reward.

Ni0( n ) :- Number of times machine ‘i’ was selected up to n times but we did not get the reward.

So, We will make a list,

Ni1( n ) as reward\_list\_0 = [0] \* 10 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

Ni0( n ) as reward\_list\_1 = [0] \* 10 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Step 2 : - For Each Round we are computing the increment Ni1( n ) and Ni0( n ) for each machine upto N rounds.



Step 3 :- Then, We are comparing all the machine’s **Beta distribution** and selecting the **Beta distribution** among all the machine for that round and keep on appending the machine in a separate list by tracking which machine is giving the highest **Beta distribution** for that round till N rounds.

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