**Adset Optimization**

# Objective

Companies have a suite of potential adsets they can display to visitors, but the company is not sure which adset strategy to follow to maximize click through rate or any other metric required as per business objective. Adset optimization is basically quantifying the value of accounting for attributes describing different adsets and unobserved differences in viewer’s responsiveness to those adset attributes. In other words, to identify what all adsets are most effective at acquiring customers and also to determine which adset to serve on which channel. The challenge then becomes how do you identify the subset of available adsets that will perform the best but at the same time be the most cost effective.

So after having found the optimal budget allocation across different channels and the target audience, the next coherent step in data driven market optimization is to maximize customer acquisition by testing many adsets on different traffic sources while learning which adset works best on which channel. To tackle this problem we have chosen to apply Multi Armed Bandit models, a multi-armed bandit solution is a reinforcement-learning algorithm that uses machine-learning algorithms to dynamically allocate traffic to adsets that are performing well, while allocating less traffic to adsets that are underperforming.

# Methodology

### Steps:

* Data ingestion, cleaning and transforming it into a standardized format.
* Identify and correct missing data points/anomalies as required
* Understand the format of the data and begin with basic exploratory data analysis (trend and outlier analysis).
* Establish a baseline model that we aim to exceed.
* Identify which algorithm of Multi Armed Bandits model gives the most optimal solution based on the kind of data availability.
* An optimization model is built to distribute the budget across different adsets with the objective of maximizing customer engagement (CTR) subject to certain constraints. These constraints will be based on the business requirements. In case of a small budget, we can build a linear optimization model to solve the problem.

# Process Flow Components

## Feature Engineering

Feature engineering is the process of using domain knowledge of the data to create features that would make the adset optimization model work.

Initially all the adsets are run randomly on a traffic source and their corresponding impressions, clicks and spend is recorded.

After that we manually define what the input variables should be and what all new variables should be created as per the requirement of the model, e.g. using the data, probability that an adset will get clicked upon its impression is calculated, i.e., CTR (click through rate) for each adset which serves as the probability that and adset will return a reward (click).

## Exploratory Data Analysis

Exploratory data analysis (EDA) is an approach to analyzing data sets to summarize their main characteristics, often with visual methods.

Here we will analyze the trends and check for the existence of outliers, if any. Outlier in the sense of adset optimization means an adset with very low impressions (exposure), which in turn depicts a misleadseting picture of customer engagement acquired by that particular adset.

Trend analysis tries to capture the pattern of performance of each adset, i.e., how the CTR of each adset is behaving, is it increasing, decreasing or is constant.

## Multi Armed Bandit

The multi-armed bandit problem is a problem in which a fixed limited set of resources must be allocated between competing (alternative) adsets in a way that maximizes their customer engagement, when each adset’s properties are only partially known at the time of allocation, and may become better understood as time passes or by allocating resources to the choice.

Multi Armed Bandit models have an added advantage of naturally minimizing strategies that do not work and not only it provides with the optimal allocation of adset impressions but simultaneously learns how to grow conversions in the future by continuously learning from the past behavior.

The algorithm starts in an ignorant state, where it knows nothing, and begins to acquire data by testing the system. As it acquires data and results, it learns what the best and worst behaviors are (in this case, it learns which adset is the best).

Furthermore, adset testing tries to solve the explore-exploit problem in a different way. Instead of two distinct periods of pure exploration and pure exploitation, bandit tests are adaptive, and simultaneously include exploration and exploitation.

This algorithm suggests that we should not discard the adsets that didn’t do well, but we should pick them at a decreasing rate as we gather confidence that there exist better adsets.

## Advantages of using MAB

* Earn While You Learn: Data collection is a cost, and bandit approach at least lets us consider these costs while running optimization projects.
* Automation: Bandits are the natural way to automate the selection optimization with machine learning, especially when applying user target – since correct a/b tests are much more complicated in that situation.
* A Changing World: By letting the bandit method always leave some chance to select the poorer performing option, you give it a chance to ‘reconsider’ the option effectiveness. It provides a working framework for swapping out low performing options with fresh options, in a continuous process.

## Insights from Multi Armed Bandit Model

* Observing one adset’s performance can suggest how similar adsets will perform.
* What percentage of the budget should be allocated to each adset?

Various Bandit Algorithms are given as follows:

* Epsilon Greedy
* Softmax
* Upper Confidence Bound (UCB)
* Thompson Sampling

### Epsilon Greedy Algorithm

The epsilon-Greedy algorithm is almost a greedy algorithm because it generally exploits the best available option, but every once in a while the epsilon-Greedy algorithm explores the other available options. The epsilon-Greedy algorithm works by randomly oscillating between purely randomized experimentation and objective of maximizing profits. When a new visitor comes to the site, the algorithm flips a coin that comes up tails with probability epsilon. If the coin comes up headsets, the algorithm is going to exploit. To exploit, the algorithm looks up the historical click through rates for the adsets and determines the adset with highest success rate in the past.

If, insteadset of coming up headsets, the coin comes up tails, the algorithm is going to explore. Since exploration involves randomly experimenting with the adsets, the algorithm needs to flip a second coin to choose between them. Unlike the first coin, we’ll assume that this second coin comes up headset 50% of the time.

## 

### Softmax Algorithm

The Softmax algorithm tries to cope with adsets differing in estimated value by explicitly incorporating information about the CTR of the available adsets into its method for choosing which adset to select when it explores.

Softmax algorithm handles this problem by imagining that you choose each arm in proportion to its estimated value. Suppose two arms, A and B, based on the past experiences have hadset two different rates of success: rA and rB (CTR). With those assumptions, the implementation of a Softmax algorithm would have you choose Arm A with probability exp(rA) / (exp(rA) + exp(rB)) and Arm B with probability exp(rB) / (exp(rA) + exp(rB)).

### Upper Confidence Bound Algorithm

Upper Confidence Bound Algorithm can be summed up by the principle of “optimism in the face of uncertainty”. That is, despite our lack of knowledge in which adsets are the best we will construct an optimistic guess as to how good the expected payoff of each adset is, and pick the action with the highest guess. If our guess is wrong, then our optimistic guess will quickly decrease and we’ll be compelled to switch to a different adset. But if we pick well, we’ll be able to exploit that action and incur little regret. In this way we balance exploration and exploitation.

Optimism comes in the form of an upper confidence bound, specifically we want to know with high probability that the true expected payoff of an adset is less than our prescribed upper bound. One general (distribution independent) way to do that is to use the Chernoff-Hoeffding inequality.

### **Thompson Sampling Algorithm**

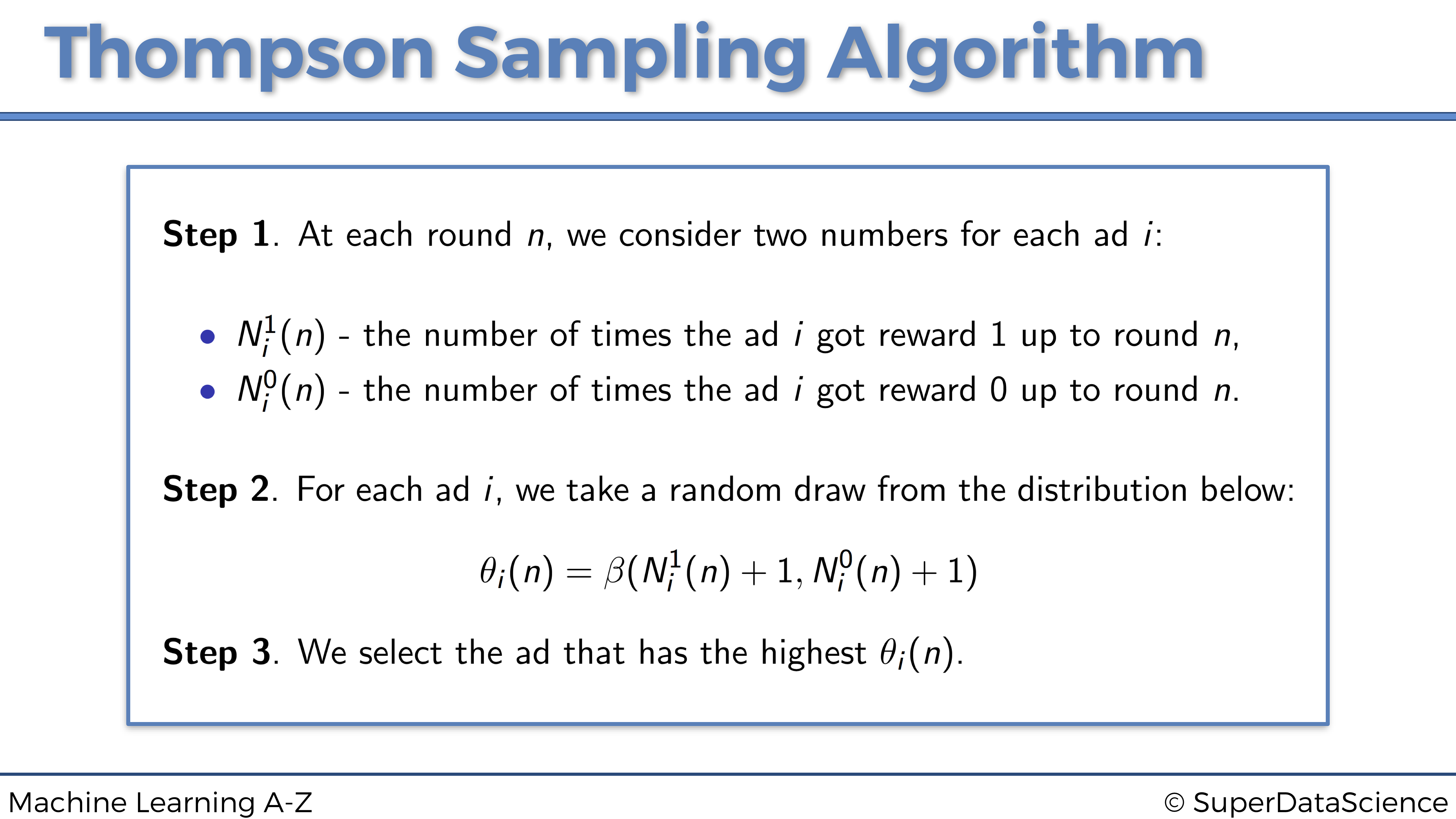
Thompson Sampling is a heuristic for choosing actions that adsetdresses the exploration-exploitation dilemma in the multi-armed bandit problem. It consists in choosing the action that maximizes the expected reward with respect to a randomly drawn belief

### Implementation

Considering that there are K number of adsets that are to be optimized,

First we play each of the K adsets randomly and find out their respective average CTRs, i.e., overline{x}_i of each adset i.

Then at every round of iterations, we consider the selection of adset derived from a betavariate distribution and then we select the ads which have the highest distribution gathered from the distribution. We then sum up the rewards and assign the ad weightage on the basis of the percentage of total weights.



Here **beta** is drawn from a random bivariate distribution.

## Selection Criteria

Algorithms of multi-armed bandit model are based on different distributions and inequalities, so our criteria towards selection of one is based on how well each algorithm is maintaining the ratio of exploration and exploitation of adsets on the data given.

## Budget Optimization

After having found the respective ratios in which adsets are to be shown, the next step is to allocate the pre decided fixed budget among these adsets. The first step is to define the objective and constraints based on the business knowledge and requirements.

Our objective is to optimize the budget allocation across various adsets while maximizing customer engagement (CTR). The constraints for this optimization could be similar to the following:

* Minimum exposure that must be met by an adset. Example: Budget should be allocated such that at least 1,000 impressions of a particular adset are shown.
* adsets are to be shown in a particular ratio i.e., if some adset has a weight 0.2 that means of all the impressions, 20% should be of that particular adset.
* Total budget constraint, i.e., how much is the company willing to spend on adsets displayed on Facebook channel.

For small or moderate media budget, Linear Programming is a viable approach in solving this problem. Linear programming is a technique for the optimization of a linear objective function, subject to linear equality and linear inequality constraints.

### Components of a linear equation

### Objective function

Function that maps an event or values of one or more variables onto a real number intuitively representing some "cost" associated with the event. In our case, we are looking to optimize our budget to maximize clicks. Our objective function would be the product of a variable representing the allocated spends for the channels and a variable representing total number of clicks per unit spend.

Mathematically the objective function can be defined as:

### Z = ∑ (Xc \* Ic)

Where Xc represents the spend across each adset and Ic represents the click through rate of each adset.

This metric is calculated based on the historic data that is available for the channel. It can be calculated using the rolling average over a defined time period or by certain time series methods that are also capable of capturing seasonality in the data.

### Equality and Inequality Constraints

These are the constraints adsetded on the variables that are involved in formulation of the model. Mathematically, we can write these as follows:

If we want a minimum number of impressions for adset1, then one of the constraint can be X(adset1)\*IPS(adset1) > α

Where α is the minimum impressions desired and IPS is the Impressions per spend which can be inferred from the historical data. X (adset1) is the spend for adsetvertisement 1.The constraints are generally defined as a part of the business scenario that is currently being dealt with.

### Implementation and Solution

The most popular method for linear optimization is the Simplex Method. adsetvantage of using a Simplex method is that it is easy to understand and implement and is also able to evaluate whether no solution actually exists.

The simplex method uses a systematic strategy to generate and test candidate vertex solutions to a linear program. That is, iteratively it chooses the variable that can make the biggest modification toward the optimum solution. It continues to solve until there is no more improvement possible without violating the user-defined constraints.

The simplex method can be implemented using the scipy library of Python that has a function linprog. Most of the calculations are abstracted underneath and they provide a simple interface to interact with.

# Overall Process

* Upon the application of algorithm we get the ratio in which the adsets should appear in the next interval (week) of run.
* Using these initial weights for each adset, along with certain other constraints from business perspective, budget in the next week is allocated across adsets using linear optimization model.
* Further, every weekly update will take into consideration the performance (CTR) of each adset in previous weekly runs along with the initial randomized run of adsets.
* A few weeks down the line, algorithm will eventually start learning and converge towards the winner adset, so it is important that this convergence does not get deviated due to ambiguity of previous results.
* To overcome this problem, CTR of each adset for every coming week will finally be calculated by taking weighted average of CTRs obtained previously where weights would be allotted on the basis of recency, more the recency, higher the weight.

# Case Study

## Data Description

We obtained data from Facebook for client A on digital adset level investment. It includes information for 242 different adsets, which were randomly shown on Facebook for the months of May-June in the year 2018. Out of the features available, we restricted the dataset to 6 variables – date, adset id, impressions, clicks, spend and conversions.

Further new variables were created using aforementioned features from analysis perspective. These variables included click through rate (CTR), clicks per spend (CLPS), conversions per spend (CPS) and impressions per spend (IPS).

## Exploratory Data Analysis

### Outlier Analysis

We defined outliers in the dataset as the adsets that hadset a low exposure, i.e., adsets where CTR has been greater than 0. Also, adsets that were not considered for the analysis are not eliminated entirely rather they can be used in future runs and after getting enough exposure, can be tested during further weekly adset optimization.

### Trend Analysis

At adset level we didn’t come across any particular trend, which means that CTR of each adset wasn’t following any pattern. While at date level, it was seen that there was an increasing trend across CTR (of all adsets combined) towards the end of each month.

## Implementation of Multi - Armed Bandit Model

Implementation of multi armed bandit model requires the probabilities of each adset getting clicked upon an impression where Click Through Rate (CTR) of each adset provides us with that probability.

Further, we ran each algorithm 373 times (iterations) having 1000 episodes each and the experiment with highest reward was chosen for analysis. By reward we mean stochastic rewards from the adsets, +1 reward for success and 0 reward for failure.

Also, since implementation of MAB happens on a real time basis, through this modeling we can only provide for the initial distribution of the adsets that will get updated after every weekly run.

**Thompson Sampling Algorithm**

In Thompson Sampling, we use Bayesian statistics in order to determine our confidence that each of the headsetlines is currently the best. We then play the headsetlines in proportion to our confidence.Conversely, when a clear winner starts to emerge, we will play that winner more and more. Companies like Google have also implemented Thompson Sampling algorithms to great success.

## Conclusion

On the application of Thompson Sampling algorithm on procured dataset, we observed that in every iteration the algorithm was converging towards the optimal adset allocation. As time progresses, based on sufficient sample size thompson sampling will converge to the optimum adset allocation as it models the underlying distribution of adset selection closely to the actual bivariate distribution of the user engagement.

## Budget Optimization

Financial planning regarding digital media marketing takes place before a company actually goes aheadset with the adsetvertising of its product, so we alreadsety have a pre-decided budget that the client wants to invest in Facebook. Now, as we know the respective weights for the adsets, consequently one simple way of allocating the budget among different adsets is to divide according to their respective weights. However, it is possible that there are adsetditional constraints that require to be fulfilled as per the business requirements, in that case we make use of linear optimization model, which not only takes care of the distribution of adsets in the required proportion but also satisfies various other constraints like minimum exposure for each adset and maximum investment per adset.

Here we define our objective function as a maximizing problem wherein we are maximizing customer engagement (clicks) subject to following constraints:

* Minimum exposure that must be met by an adset, i.e., budget should be allocated such that at least 1,00 impressions of a particular adset are shown.
* Adsets are to be shown in a particular ratio i.e., if some adset has a weight 0.2 that means of all the impressions, 20% should be of that particular adset.
* Total budget constraint, i.e., budget fixed for Facebook.
* Individual adset budget constraint, i.e., maximum spend for each adset.

Mathematically, the problem can be formulated as follows:

Maximise, ***z =***

subject to,

, *∀* *i*

where, : Click through rate of the *ith* adset   
 : Spend allocated to the *ith* adset   
 *C* : Total budget  
 *D*  : pre - defined maximum value of spend for each adset  
 : weights obtained for the *ith* adset

On the application of this model on our data set we obtained the respective optimal clicks, impressions and spend and compared the parameters with the results of actual allocation. Upon assessment it was found that adset company allotted budget across adsets as per the weights indicated by Multi armed bandit algorithm , there would have been a 8.82% increase in customer engagement.

# References

<https://jeremykun.com/2013/10/28/optimism-in-the-face-of-uncertainty-the-ucb1-algorithm/>

<https://conversionxl.com/blog/bandit-tests/>

[https://books.google.co.in/books?id=xnAZLjqGybwC&printsec=frontcover#v=onepage&q&f=false](https://books.google.co.in/books?id=xnAZLjqGybwC&printsec=frontcover" \l "v=onepage&q&f=false)

[https://imadsetdabbura.github.io/blog/data%20science/2018/03/31/epsilon-Greedy-Algorithm.html](https://imaddabbura.github.io/blog/data science/2018/03/31/epsilon-Greedy-Algorithm.html)

<http://www.stephacking.com/upper-confidence-bound-ucb-python-implementation/>

<http://www.stephacking.com/upper-confidence-bound-ucb-python-implementation/>

<https://sudeepraja.github.io/PureExp/>

<https://www.quora.com/What-is-Thompson-sampling-in-laymans-terms>

[https://github.com/lilianweng/multi-armed-bandit/blob/master/solvers.py#L76](https://github.com/lilianweng/multi-armed-bandit/blob/master/solvers.py" \l "L76)

<https://github.com/ankonzoid/Deep-Reinforcement-Learning-Tutorials/blob/master/multiarmed_bandit/multiarmed_bandit.py>

<https://conversionxl.com/blog/bandit-tests/>

[http://www.pablomaldonadseto.org/regret-minimization-in-python/](http://www.pablomaldonado.org/regret-minimization-in-python/) <https://www.chrisstucchio.com/blog/2012/bandit_algorithms_vs_ab.html> <https://itnext.io/reinforcement-learning-with-multi-arm-bandit-decf442e02d2>