Pybandit: A Website Optimization Framework for E-commerce SMBs

Abir Das
Tuhin Sharma

Who we are

MSc(OR) Student
@DU

Sr Principal Data Scientist
@RedHat

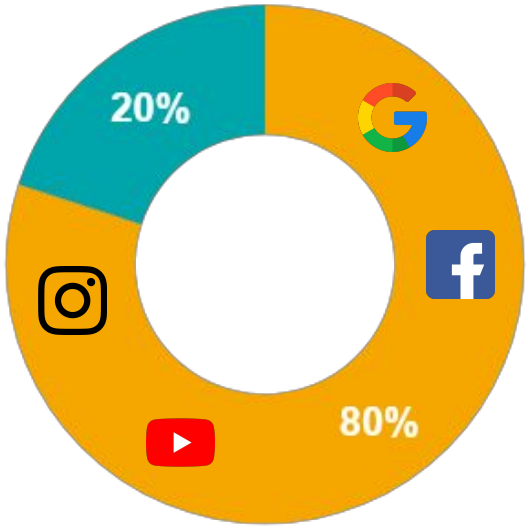
Agenda

- Problem statement
- Big Company vs SMBs
- Existing Approach (AB Testing)
- Problems with existing approach
- Pybandit Approach (Multi Armed Bandit)
- AB Testing vs Multi Armed Bandit
- Advantages of Pybandit
- Demo

Problem statement

Merchants, with the help of their digital partners, know how to get traffic to their online shop

Digital Marketing Spend



Facebook, Instagram and Google have sophisticated algorithms to help merchants get the most relevant traffic for their ads

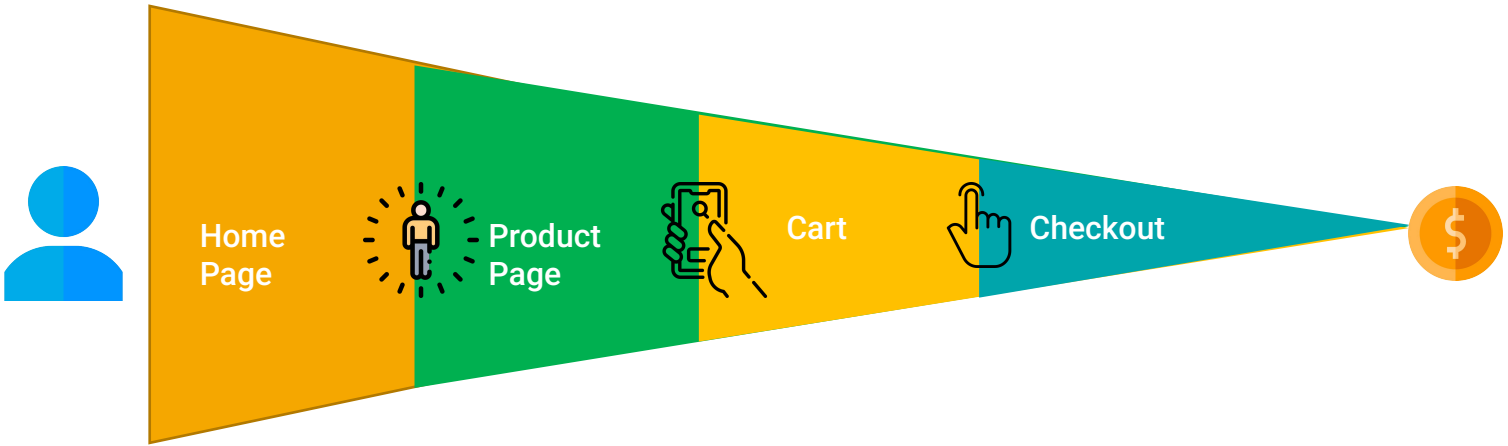
Offline vs Online

Brick & Mortar conversions > 20%

But eCommerce conversions < 2%

Offline stores have greater conversion rate than online counterpart

This happens because the website is made to work harder and not smarter.



Website, is not refined and updated regularly. → Ineffective store → Very low conversion

The situation has worsened as well as become extremely critical in the current pandemic era

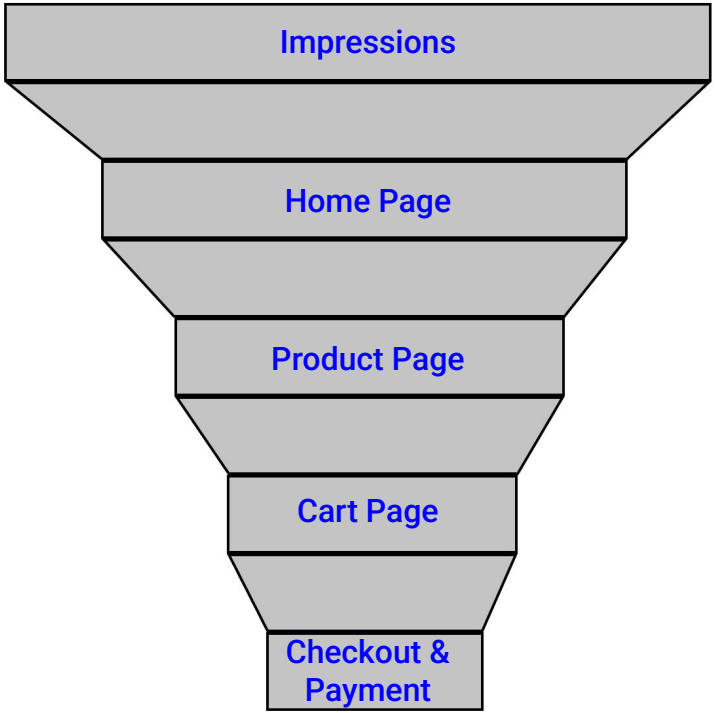
Consumers have rapidly embraced digital

Both brick & mortar and digital-native companies are adjusting to eCommerce reality

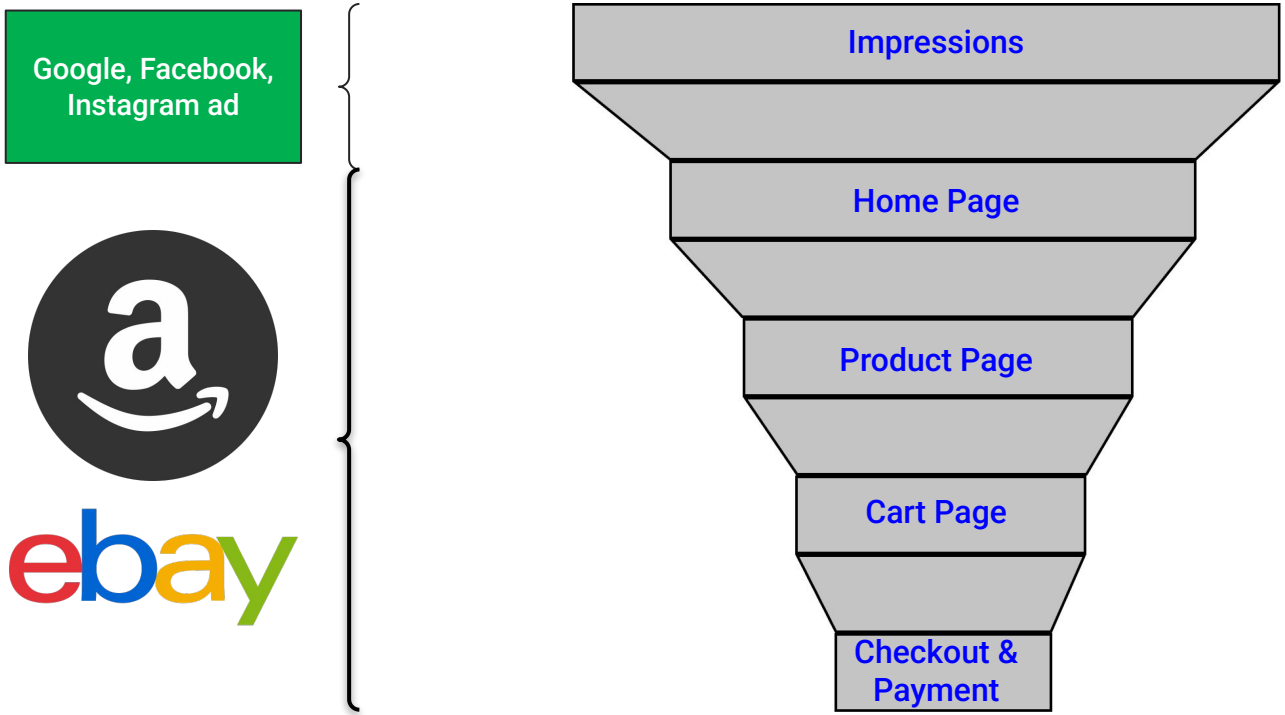
Marketing budget is squeezed

Advantages of Big Companies

Customer Funnel



Customer Funnel: Big Ecommerce Business

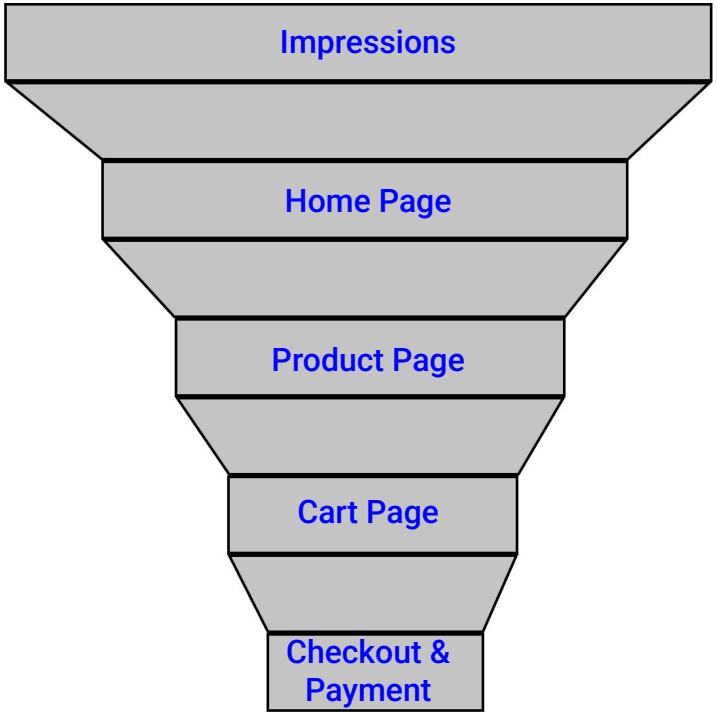


Customer Funnel: Small & Medium Ecommerce Business

Google, Facebook, Instagram ad

GAMBOL
LET'S SPOT A SPORT IN YOU

Shopify, Bigcommerce etc.



Big Business vs Small & Medium Business

Big Companies have in-house teams of Data Scientists, ML engineers to enhance conversion

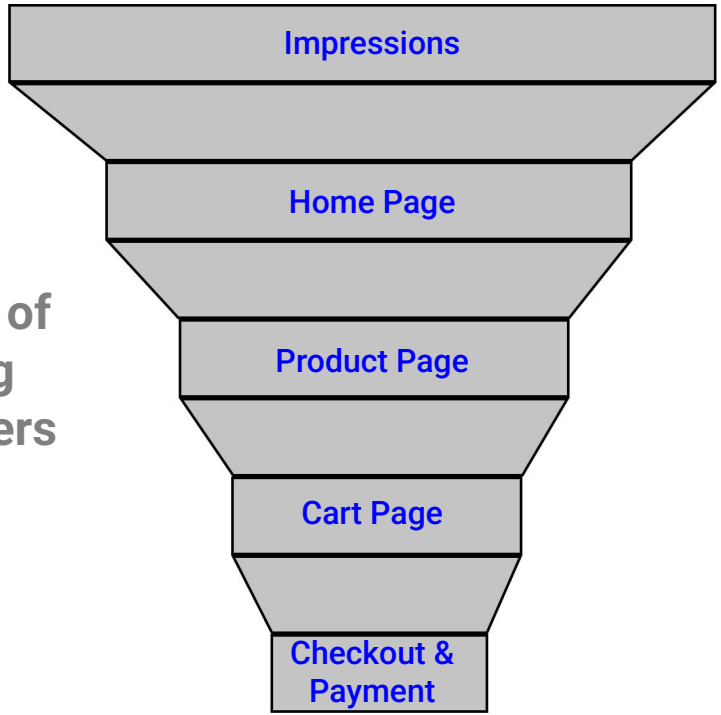
Pybandit is here to rescue

Google, Facebook, Instagram ad

Pybandit

Shopify, Bigcommerce etc.

focuses on the middle of the customer shopping journey for SMB retailers



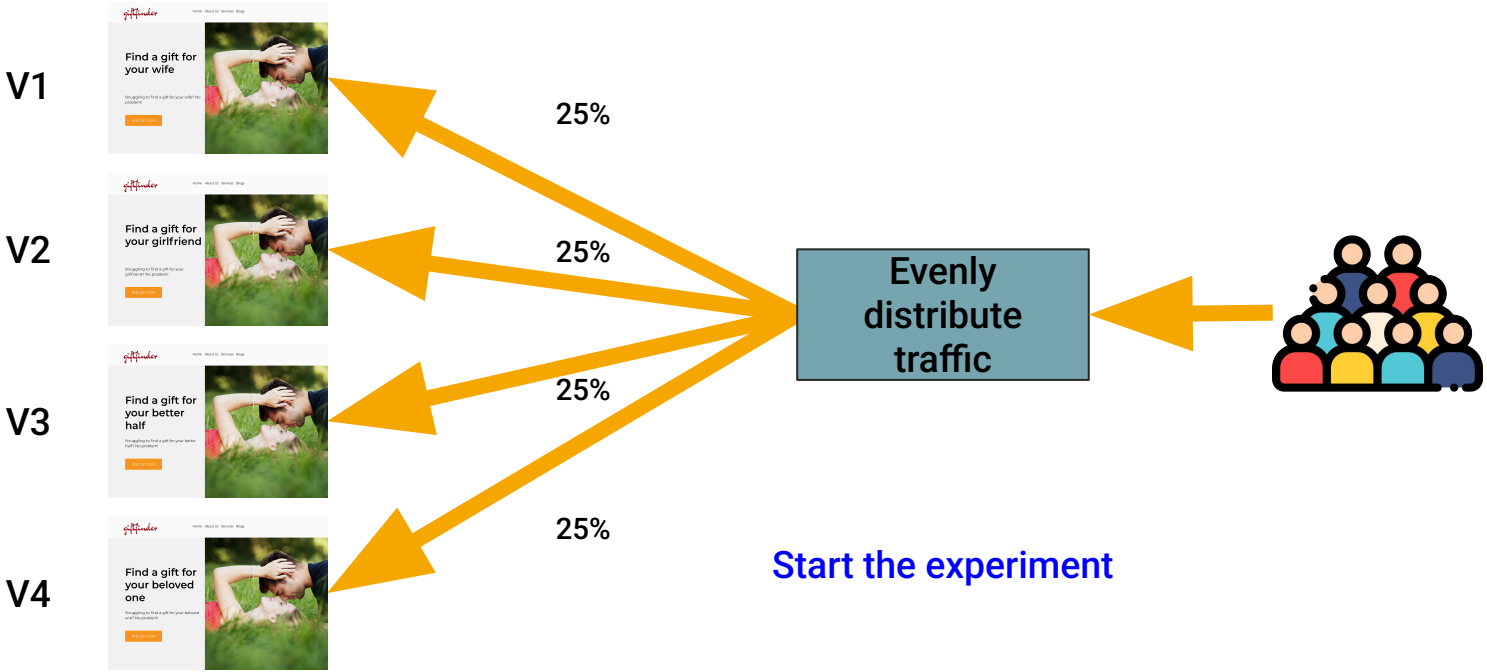
Pybandit is here to rescue

Pybandit helps you optimize

- Design
- Messaging
- Pricing

Existing Approach (AB Testing)

Existing Approach Step 1 : Split



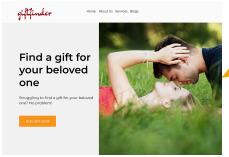
Existing Approach Step 2 : Explore

V1

V2

V3

V4



25%

25%

25%

25%

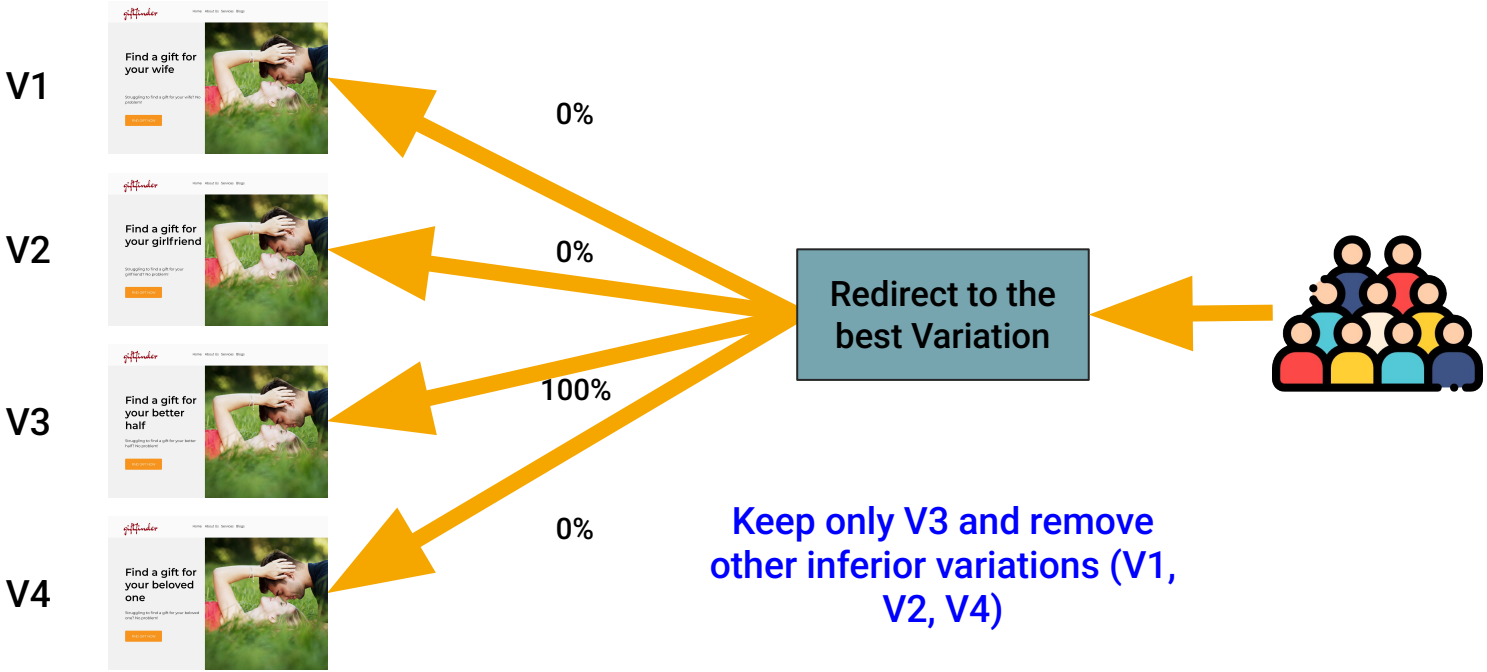
Run the process for 3 weeks

Evenly distribute traffic

Winner!!

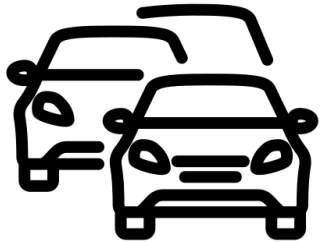
Variation	Visitors	%total traffic	Conversion	%conversion
V1	20840	25	630	3.02
V2	20660	24.8	660	3.18
V3	20760	24.9	920	4.41
V4	20800	25	580	2.79

Existing Approach Step 3: Exploit



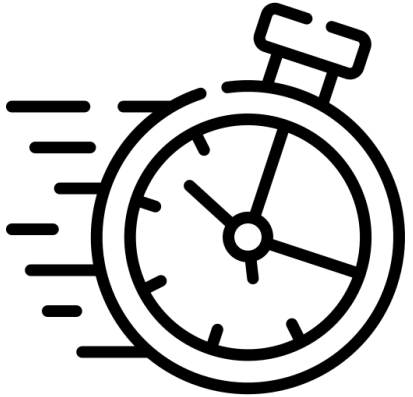
Disadvantages of Existing Approach

AB Testing is not ideal for SMBs



Low traffic, hence longer time to conclude.

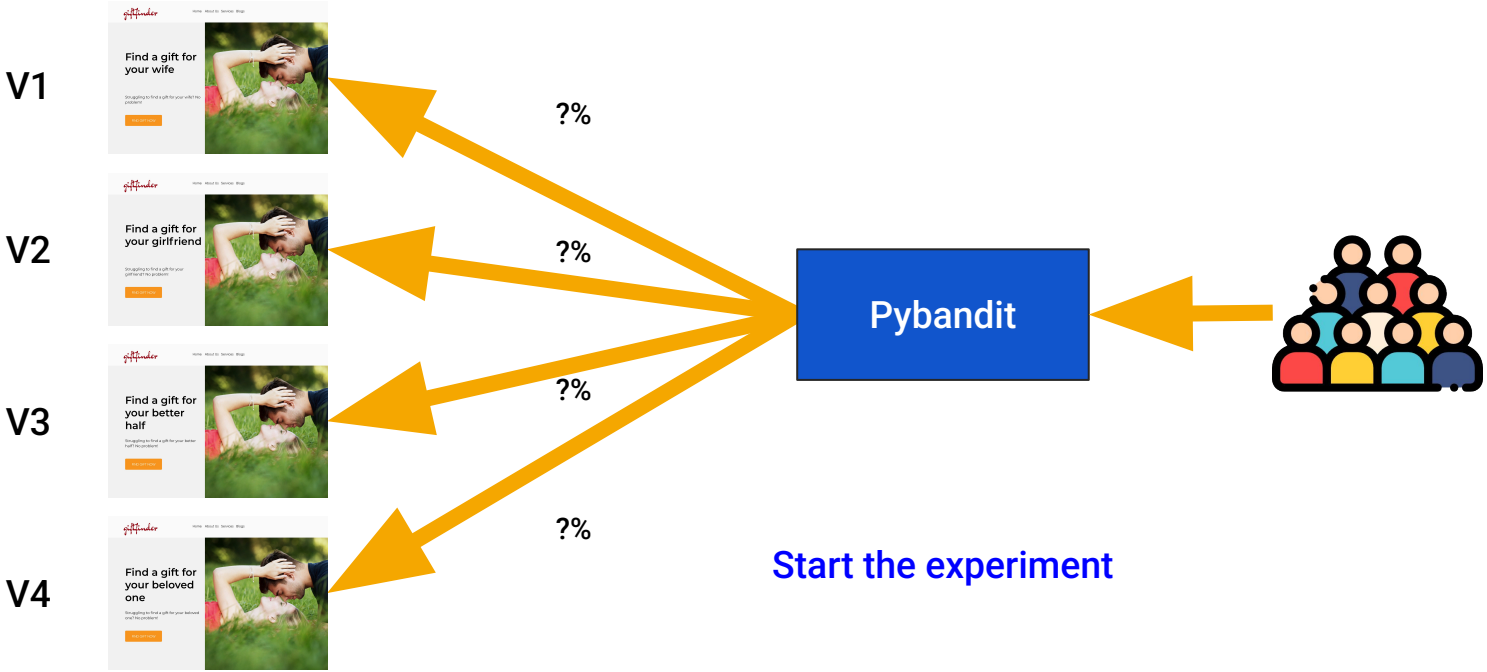
Fear of missing out as wrong variations are dominant



Fast moving business can’t afford sequential experiments

Pybandit Approach (Multi Armed Bandit)

Pybandit Approach Step 1 : Realtime Split

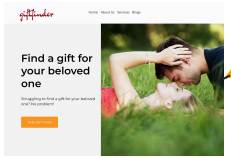


Pybandit Approach Step 2 - Explore with Feedback

V1

V2

V3



V4

8%

9%

76%

7%

Pybandit

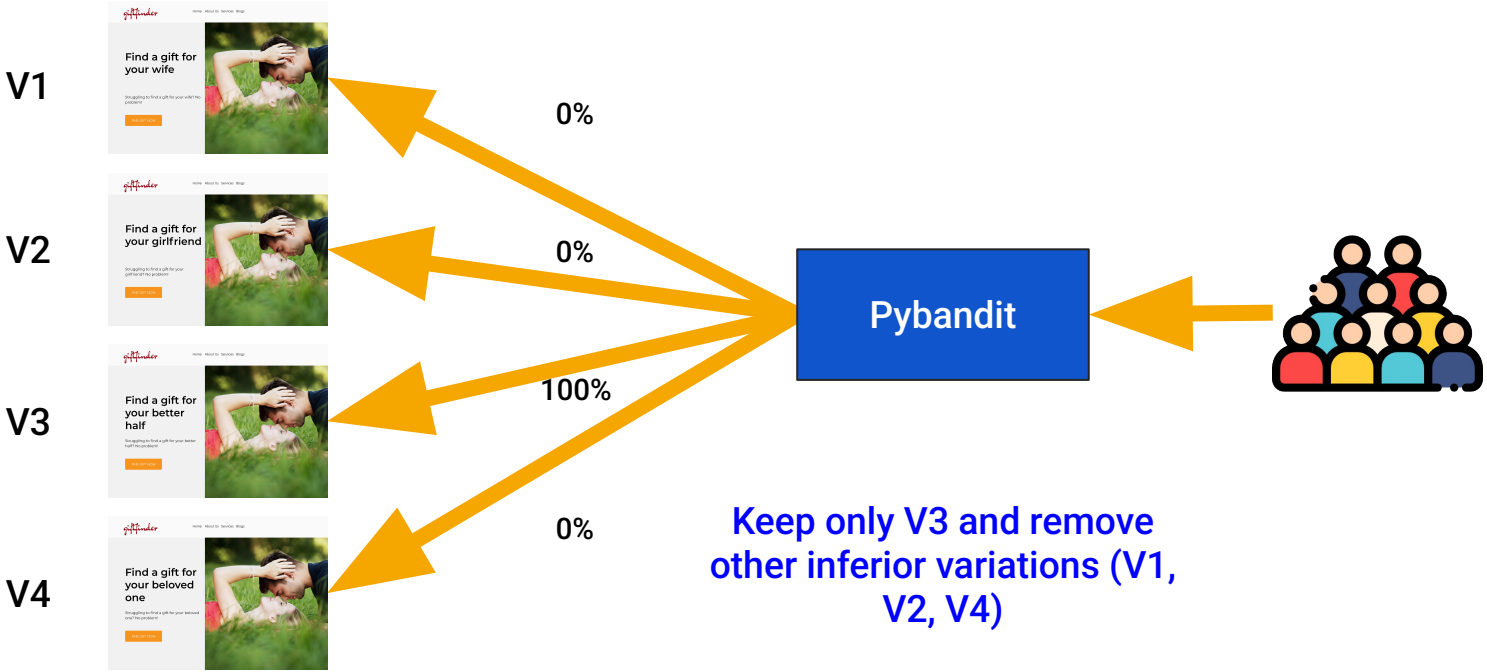
Real-time Feedback loop powered by the current performance of variations at that moment in time

Variation	Visitors	%total traffic	Conversion	%conversion
V1	6400	7.7	220	2.99
V2	7220	8.6	200	3.12
V3	63220	76.1	2740	4.32
V4	6220	7.4	180	2.82

Winner!!

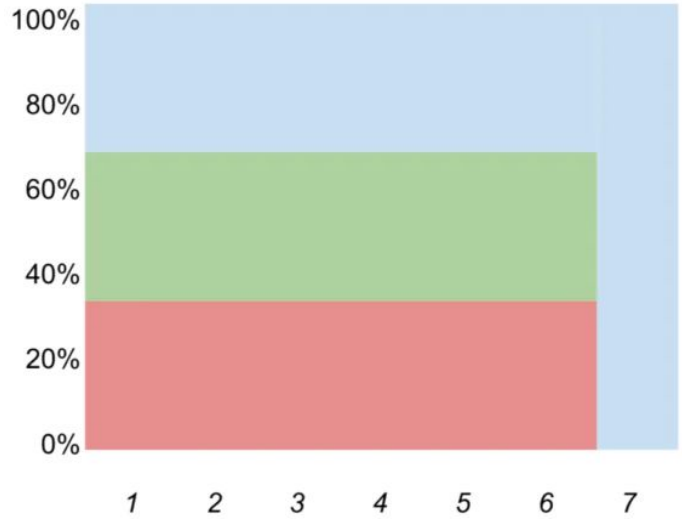
Run the process for 3 weeks

Pybandit Approach Step 3: Exploit Automatically

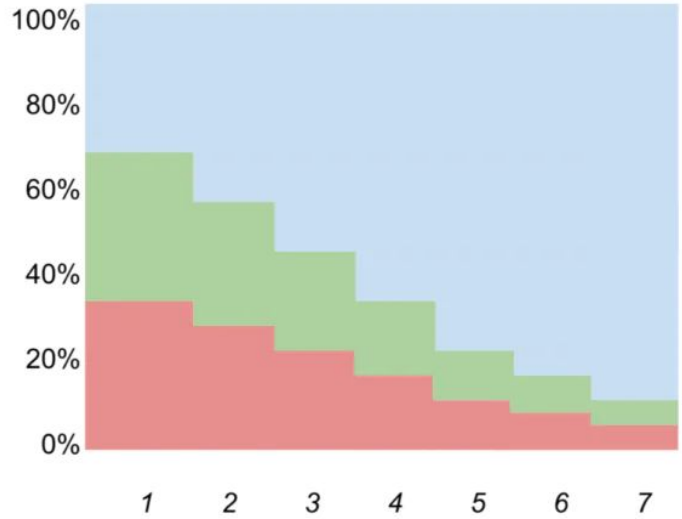


AB Testing vs Multi Armed Bandit

Pybandit maximizes revenue during exploration



AB Testing
(Exploit follows Explore)



MAB
(Explore and Exploit happens simultaneously)

Advantages of Pybandit

Pybandit maximizes revenue during exploration

Existing Approach

Variation Type	Variation	# Impressions	% of traffic served
best	V3	20760	25
worse	V1, V2, V4	62300	75

Pybandit Approach

Variation Type	Variation	# Impressions	% of traffic served
best	V3	> 58140	> 70
worse	V1, V2, V4	< 24920	< 30

	Conversion	% Conversion
Existing Approach	2790	3.35
Pybandit Approach	3330	4.0

- Increase in Conversion ~ 20%
- Each conversion - $100,
  Increase in Revenue = $ (3330-2790) * 100 = $54,000

For the whole Exploration phase (3 Weeks)

Pybandit auto-detects completion of exploration

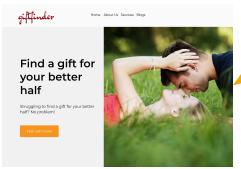
	Detect Completion of Exploration	**Switch to best variation**
Existing Approach	Human intervention needed	Takes 4-5 business working days for the development team
Pybandit Approach	Automatic Notification	Automatic Redirection

Pybandit supports Dynamic Multivariate Testing

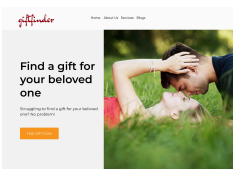
V1

V2

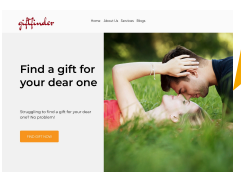
V3



V4



V5



Remove V4 on the 5th day in the exploration phase

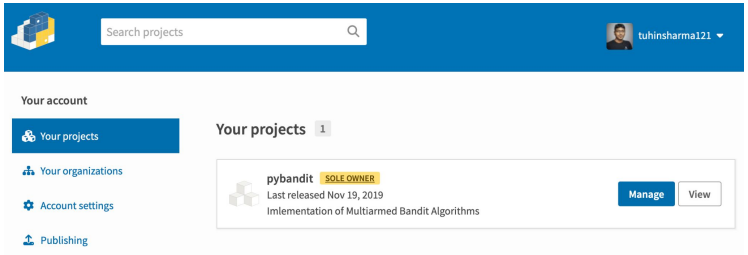
Pybandit

Add new variation V5 added on the 10th day in the exploration phase

	Process
Existing Approach	• Scrape the result • Restart the experiment
Pybandit Approach	• Add/remove anytime. • Experiment continues.

Demo - Pybandit Library

Pybandit is in active development and available in Pypi



https://github.com/tuhinsharma121/pybandit

pybandit

A Python library for all popular multiarmed bandit algorithms.

Roadmap

The following algorithms are currently being explored before we set a roadmap for the first release.

1. Epsilon Bandit:
    i. Epsilon-greedy strategy (done)
    ii. Epsilon-first strategy (in progress)
    iii. Epsilon-decreasing strategy
    iv. Epsilon-adaptive strategy
2. Bayesian Bandit
    i. Thompson Sampling
3. Contextual Bandit
    i. Linear Classifier
        a. LinUCB (Upper Confidence Bound) algorithm:
        b. LinRel (Linear Associative Reinforcement Learning) algorithm:
    ii. Non-linear Classifier
        a. UCBogram algorithm
        b. NeuralBandit algorithm
        c. KernelUCB algorithm
        d. Bandit Forest algorithm
    iii. Constrained
        a. UCB-ALP algorithm
    iv. Greedy
        a. Contextual-Epsilon-greedy strategy
4. Adverserial Bandit
5. Dueling Bandit
6. Collaborative Bandit
    i. COFIBA

Thanks