Greatplant XP: Changing the layout of landing page will have what impact on leads signing-up

Last updated: Nov 6th, 2024   Author: Zicheng Bu

Summary

1. **Goal of the XP:** Product team refines the signup process and would like to see this change will impact leads signup in which way.

2. **Timeline of the XP:**
   a) Planning: Oct 22th, 2024
   b) Rollout: Oct 23th, 2024 00:00:00 - Nov 5th, 2024 23:59:59
   c) Analysis: Nov 6th, 2024

3. **Unit of Randomization: User level.**
   a) Control group: 598
   b) Treatment group: 603

4. **Metrics of interest:**
   a) **The primary metric: average numbers of sign-up leads** has an **significant increase** of 0.1239.
   b) **The secondary metric sign-up leads and conversion rate from non-signups to signups** has an **significant increase** of 10.65 pp.
   c) Intrepretation: The new design successfully increase the number of signups and conversion rate:

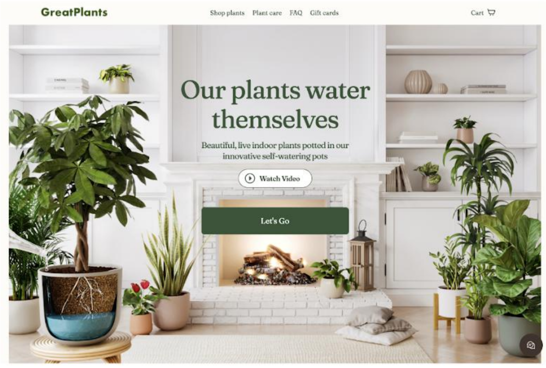
5. **Guardrail Metrics:**
   a) **No significant negative effect on the guardrail metrics:** average number of orders and average total order prices.
   b) Interpretation: The leads show consistent purchasing pattern.

6. We suggested rolling out the feature.

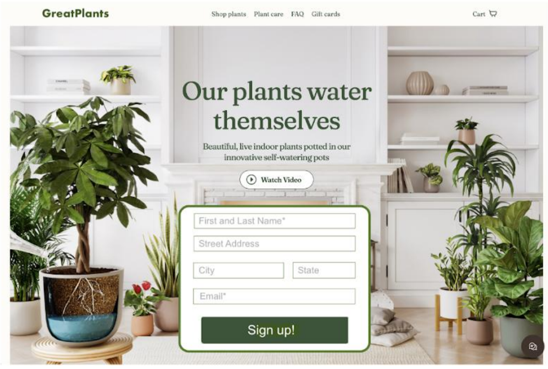
Time: Oct 22th, 2023
Planning phase

1. Description of the feature and experiment:

Product team wants to change the layout of landing page. The idea is that the previous signup process(shown in the left) takes several pages and much time to finish. By contrast, new design(shown in the right) simplifies the process by showing all information to fill out at the landing page, which could probably save much time, lower the cognitive load of signing up, and result in **more leads to sign up**.


GreatPlants
Shop plants
Plant care
FAQ
Gift cards
Cart
Our plants water themselves
Beautiful, live indoor plants potted in our innovative self-watering pots
Watch Video
Let's Go

Control



Treatment

2. The experiment is a classic AB experiment:

a) For a period of two week a certain number of users (defined below in the planning section) will be split 50/50 into 2 groups: Control and Treatment. The Control group will have the usual experience with old version.

b) After two week we will analyze the data and we will determine that either: The new feature has a significant impact on key performance metrics and should be rolled-out. Or the new feature does not show a significant improvement. In this case, we will not roll-out the new feature.

3. Experiment planning:

a) Platform: The new feature will appear (for leads in Treatment group) in every landing page on our website. No other platforms will be affected.

b) **Unit of randomization: The experiment will be administered at the user level.** It is reasonable to assume that leads behave independently from each other on our platform, and there will be no contamination between the Control group and the Treatment group.

c) The reason why we do not administer the experiment at session level is that each lead may have several sessions which are internally correlated. When calculated the ratio metric, converting session level metrics to individual level sessions may lead to misallocation of weight.

d) Control/ Treatment Group: Total number of each group will be calculated in the power analysis section below. And all enrolled leads will be randomly assigned to Control or Treatment group by 50/50.

a) Metrics: We designed 4 metrics (1 primary, 1 secondary, and 2 guardrail metrics) at user level within 2-week experiment time (Oct 23th, 2024 00:00:00 - Nov 5th, 2024 23:59:59). Details are shown below: (Default values are derived with queries in the appendix)

Metrics	**Type**	**Priority**	**Default**	**Desired Effect**
Average number of sign-up leads	Sample Mean	Primary	0.455	Increase
Conversion from non-signup leads to signup leads	Ratio Metrics: Average orders/average sessions	Secondary	25.18%	Increase

Average number of orders	Sample Mean	Guardrail	0.59	No Effect
Average total price of orders	Sample Mean	Guardrail	70.73	No Effect

4. Power Analysis:

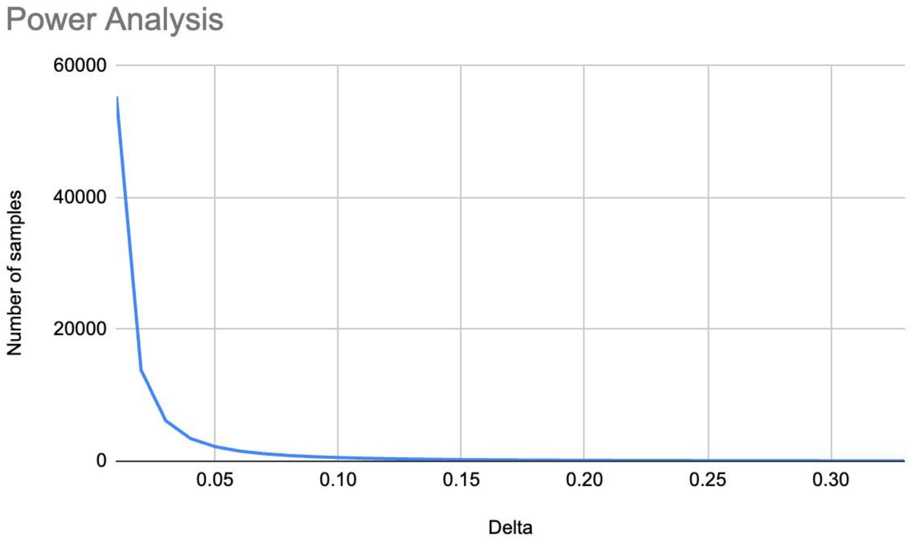
We want to run the experiment for 2 weeks (betweeen 2024-10-23 and 2024-11-5).

In the plot below:

Delta (on the y-axis) represents the detectable difference that we will observe between the average number of "cart add" events in Control and Treatment.

n (on the x-axis) represents the number of users that we need in each group to detect the corresponding significant delta, after fixing alpha and beta. It was computed using the following formula:

$$n = 2\left(\hat{\sigma}\ \frac{z_{\alpha/2} + z_{\beta}}{\delta}\right)^2$$



After discussing with the Product team, the delta was set to be **22% of the mean of the primary metric.** The primary metric (Average number of signups) statistics is:

mean: 0.455
standard error: 0.4982

Fixing the alpha (probability of Type I error) to be **5%** and beta (probability of Type II error) to be **20%**, the corresponding desired sample size is 552. To make the conclusion robust, we plan to have 600 leads in each of the groups. (See calculation here: https://docs.google.com/spreadsheets/d/13iQ7X7OWBN0GmO28Zon94q5Gpxd-cTJG4aQ1PXf-Egc/edit?gid=0#gid=0)

5. Data Logging:
Enrolled leads' information will automatically be logged into:
fabriziopublic.greatplants.experiment_groups (one row per unit of any experiment)

a) experiment_name: the unique identifier of an experiment
b) unit_id: the identifier of the unit of randomization that was used in the experiment
c) grouped_at: timestamp of when a unit was assigned to a group in an experiment
d) group: the name of the group that a unit was assigned to

For every leads who start sessions on landing page their information will be logged into fabriziopublic.greatplants.dim_lead (one row per lead that ever started a session on a landing page)

a) lead_id: the lead identifier
b) lead_source: can be one of direct / search / ads / social
c) first_session_at: timestamp of the first session of a lead
d) signup_at: timestamp of the signup completion. If NULL, the lead never signed up.
e) lead_info: encrypted information provided by the lead during signup (name, address, email, etc). If NULL, the lead never signed up.

For every session on the landing page, data is logged into fabriziopublic.greatplants.fact_landing (one row per session that started on a landing page)

a) lead_id: the lead identifier
b) session_id: the session identifier
c) session_at: timestamp of the session
d) signup_at: if a signup was completed during the session, the timestamp of the signup is recorded here.
e) order_id: if an order was placed during the session, the order_id is recorded here.

For every order, data is logged into fabriziopublic.greatplants.fact_order (one row per order, which could be generated from a session that started on a landing page or from a different flow)

a) lead_id: the lead or client identifier (even after a lead signs up, we still identify them with their lead_id)
b) order_id: the order identifier
c) order_at: timestamp of the order
d) tot_price: total price of the order, before taxes
e) taxes: sales tax applied to the order
f) tot_cost: total cost paid by the customer, including taxes

Time: Oct 23rd, 2023
Rolling out phase

1. On 2024-10-23 everything was in place to start the experiment. We used this day to start with a slow roll-out (1% and 10%) and monitored the systems. Everything was ok. In particular the allocation of users in Control and Treatment was ~50/50.
2. On 2024-10-24 we officially rolled-out to our 1200 users (100%), which started to enter in the experiment. The platform automatically logs the corresponding data.

Time: Nov 6th, 2024
Analysis phase

Time span: "2024-10-23 00:00:00 UTC" - "2024-11-05 23:59:59 UTC" (included)
Number of leads:

Control Group: 598
Treatment Group: 603

We don't see exact 600 users in each group because of some minor spillage (data that was not logged correctly). The 50/50 randomization has worked well since we see almost exactly 50% in Control and 50% in Treatment.

Given that we will perform 4 tests, we will use the Bonferroni correction to declare a result statistically significant. That is, instead of setting a threshold of 0.05 for our p-values, the threshold is 0.05/4 = **0.00125**. And the test outcomes are shown below. (Calculation is in the google sheet: https://docs.google.com/spreadsheets/d/13iQ7X7OWBN0GmO28Zon94q5Gpxd-cTJG4aQ1PXf-Egc/edit?gid=0#gid=0)

	Control	**Treatment**	**Diff T-C**	**p-value**
Average number of sign-up leads	0.4565	0.5804	0.1239	0.000015
Conversion from non-signup leads to signup leads	0.2521	0.3586	0.1065	0.00000032
Average number of orders	0.6538	0.5903	-0.0635	0.1148
Average total price of orders	78.557	71.1381	-7.4189	0.1271

Conclusion

There are significant (under alpha = 1.25% after Bonferroni Correction) positive effect on primary and secondary metrics: average numbers of sign-up leads and conversion rate from non-signups to signups, which shows **the new design successfully increase the number of signups and conversion rate**:

a) average numbers of sign-up leads has an increase of 0.1239 with p-value of 0.000015

b) sign-up leads and conversion rate from non-signups to signups has an increase of 10.65 pp with a p_value of 0.0000003.

Moreover, **the Guardrail metrics:** Average number of orders and Average total price of orders **show no significant difference,** which shows **the leads show consistent purchasing pattern.**

Therefore we recommend to roll out the new feature to all users.

Appendix

1. Default Value Metrics 1:

```
with id as (
 select lead_id, count(signup_at) as num
 from fabriziopublic.greatplants.fact_landing
 where session_at >= CAST("2024-10-9 00:00:00 UTC" as TIMESTAMP)
 and session_at <= CAST("2024-10-22 23:59:59 UTC" as TIMESTAMP)
 group by lead_id
)
select avg(num) as ave, stddev_samp(num) as std
from id
```

2. Default Value Metrics 2:

```
with id as (
 select lead_id, count(*) as num_session, count(signup_at) as num_signup
 from fabriziopublic.greatplants.fact_landing
 where session_at >= CAST("2024-10-9 00:00:00 UTC" as TIMESTAMP)
 and session_at <= CAST("2024-10-22 23:59:59 UTC" as TIMESTAMP)
 group by lead_id
)
select avg(num_session) as ave_num_session, avg(num_signup) as
ave_num_signup
from id
```

3. Default Value Metrics 3:

```
with id as (
 select lead_id, count(order_id) as num_order, count(*) as num_session,
count(signup_at) as num_signup
 from fabriziopublic.greatplants.fact_landing
 where session_at >= CAST("2024-10-9 00:00:00 UTC" as TIMESTAMP)
 and session_at <= CAST("2024-10-22 23:59:59 UTC" as TIMESTAMP)
 group by lead_id
)
select avg(num_order)as ave_order
```

```
from id
```

4. Default Value Metrics 4:

```
with id as (
 select lead_id, order_id
 from fabriziopublic.greatplants.fact_landing
 where session_at >= CAST("2024-10-9 00:00:00 UTC" as TIMESTAMP)
 and session_at <= CAST("2024-10-22 23:59:59 UTC" as TIMESTAMP)
),
tot as(
 select i.lead_id, i.order_id, o.tot_price
 from fabriziopublic.greatplants.fact_order o
 right join id i
 on i.order_id = o.order_id
 ),
ind as(
 select lead_id, sum(coalesce(tot_price,0))as tot_p
 from tot
 group by 1
)
select avg(coalesce(tot_p,0)) as ave_tot_price
from ind
```

5. Metrics 1: Average number of sign_up leads

```
with ex as(
 select unit_id, e.group
 from fabriziopublic.greatplants.experiment_groups e
 where experiment_name = 'landing_2024'
),
signup as (
 select l.lead_id, l.signup_at, e.group, (case when l.signup_at is not null then 1
else 0 end) as bi_sign_up
 from  ex e
 left join fabriziopublic.greatplants.dim_lead l
 on l.lead_id = e.unit_id
),
check as (
 select *
 from signup
 where 1=1
 and signup_at is null
 or (signup_at >= CAST("2024-10-23 00:00:00 UTC" as TIMESTAMP) and
signup_at <= CAST("2024-11-05 23:59:59 UTC" as TIMESTAMP))
)
select c.group, COUNT(c.lead_id) as n_leads, avg(bi_sign_up) as avg_signup,
STDDEV_SAMP(bi_sign_up) as sigma_hat
from check c
group by c.group
```

6. Metrics2: conversion rate from initial non-signup leads to signup leads

```
with ex as(
 select unit_id, e.group
 from fabriziopublic.greatplants.experiment_groups e
 where experiment_name = 'landing_2024'
),
session as (
 select l.lead_id, l.signup_at, l.session_id, e.group, (case when l.signup_at is
not null then 1 else 0 end) as bi_sign_up,
 from  ex e
 left join fabriziopublic.greatplants.fact_landing l
 on l.lead_id = e.unit_id
 where 1=1
 and session_at >= CAST("2024-10-23 00:00:00 UTC" as TIMESTAMP)
 and session_at <= CAST("2024-11-05 23:59:59 UTC" as TIMESTAMP)
),

check as (
 select s.lead_id, max(s.group) as group1, count(distinct(s.session_id)) as
num_session, max(s.bi_sign_up) as num_sign_up
from session s
group by s.lead_id
)

select group1, COUNT(lead_id) as n_leads, avg(num_session) as ave_session,
avg(num_sign_up) as ave_signup, STDDEV_SAMP(num_session) as
std_sessions, STDDEV_SAMP(num_sign_up) as std_signups,
COVAR_SAMP(num_session, num_sign_up) as cov
from check
group by group1
```

7. Metrics 3: Average number of orders from leads through landing page

```
with ex as(
 select unit_id, e.group
 from fabriziopublic.greatplants.experiment_groups e
 where experiment_name = 'landing_2024'
),
orders as (
 select o.lead_id, o.session_id, o.order_id
 from fabriziopublic.greatplants.fact_landing o
 where 1=1
 and session_at >= CAST("2024-10-23 00:00:00 UTC" as TIMESTAMP)
 and session_at <= CAST("2024-11-05 23:59:59 UTC" as TIMESTAMP)
),
check as (
 select e.unit_id, max(e.group) as group1, count(distinct(o.order_id)) as
num_order
 from orders o
 right join ex e
 on o.lead_id = e.unit_id
 group by e.unit_id
)

select group1, COUNT(unit_id) as n_leads, avg(num_order) as
ave_num_order, STDDEV_SAMP(num_order) as std_num_order
from check
group by group1
```

8. Metrics 4: Average total amount of orders from leads through landing page

```
with ex as(
 select unit_id, e.group
 from fabriziopublic.greatplants.experiment_groups e
 where experiment_name = 'landing_2024'
),
orders as (
 select o.lead_id, o.session_id, o.order_id
 from fabriziopublic.greatplants.fact_landing o
 where 1=1
 and session_at >= CAST("2024-10-23 00:00:00 UTC" as TIMESTAMP)
 and session_at <= CAST("2024-11-05 23:59:59 UTC" as TIMESTAMP)
),
check as (
 select e.unit_id, max(e.group) as group1, count(distinct(o.order_id)) as
num_order
 from orders o
 right join ex e
 on o.lead_id = e.unit_id
 group by e.unit_id
)

select group1, COUNT(unit_id) as n_leads, avg(num_order) as
ave_num_order, STDDEV_SAMP(num_order) as std_num_order
from check
group by group1
```