

WEBSITE OPTIMIZATION THROUGH TRAFFIC AND CONVERSION ANALYSIS

I recently worked on a project to analyze and optimize an e-commerce website. I explored traffic sources, bounce rates, top-performing content, and the conversion funnel to understand user behavior better. I also used A/B testing and bid optimization for business improvement. The goal was to enhance user engagement and increase conversions through data-driven decisions.

About the data:

In this project, I worked with multiple datasets to understand website performance, user behavior, and sales. The key tables include:

1. **website_sessions** – Contains information about each user's session on the website, such as session ID, user ID, date, utm source, utm campaign, device type and http referrer. This helps in analyzing how users navigate the website and where they come from.
2. **orders** – Stores details of all orders placed on the website, including order ID, session ID, order date, user id, number of items purchased and total value. This table is used to track sales performance and revenue trends.
3. **website_pageviews** – Includes data about pages viewed by users during their sessions, like page URL, time, and session ID. It helps analyze which pages attract the most attention and where users drop off.
4. **products** – Contains product details such as product ID, name and time. This is used to analyze which products are popular and drive sales.
5. **order_items** – Stores information about individual items within each order, including product ID, order ID and price. This helps analyze product-level sales performance.
6. **order_item_refunds** – Contains details about refunded items, including product ID, quantity refunded, order ID and refund date. This is useful to understand returns and refund patterns.

```
USE mavenfuzzyfactory;

SELECT * FROM website_sessions;

SELECT * FROM orders;

SELECT * FROM website_pageviews;

SELECT * FROM products;

SELECT * FROM orders_items;

SELECT * FROM orders_item_refund;
```

1) Analyzing Traffic source:

- 1) I analyzed the top traffic sources to identify where most of the website sessions were coming from (Date 2012-03-19 to 2012-04-12).

```
#top result of traffic source
SELECT
    utm_source,
    utm_campaign,
    http_referer,
    count(DISTINCT website_session_id) AS session
FROM
    website_sessions
WHERE
    created_at < '2012-04-12'
GROUP BY
    utm_source,utm_campaign,http_referer
ORDER BY
    session DESC;
```

OUTPUT:

	utm_source	utm_campaign	http_referer	session
▶	gsearch	nonbrand	https://www.gsearch.com	3613
	NULL	NULL	NULL	28
	NULL	NULL	https://www.gsearch.com	27
	gsearch	brand	https://www.gsearch.com	26
	NULL	NULL	https://www.bsearch.com	7
	bsearch	brand	https://www.bsearch.com	7

Result:

Most of the website visitors came from Google Search, mainly through non-brand campaigns. A smaller number of visitors came from brand campaigns and Bing Search. So, I continued my analysis focusing on gsearch non-brand traffic.

- 2) I calculated the conversion rate from sessions to orders for the gsearch non-brand campaign to see how effectively those visits turned into purchases.**

```
SELECT
    count(s.website_session_id) AS session,
    count(order_id) AS orders,
    count(order_id)/count(s.website_session_id)*100 AS cnvrsn_rt_sesn_ordr
FROM
    website_sessions AS s
LEFT JOIN
    orders AS o
ON s.website_session_id=o.website_session_id
WHERE
    utm_source='gsearch' AND utm_campaign='nonbrand' AND
    s.created_at<'2012-04-14';
```

OUTPUT:

	session	orders	cnvrsn_rt_sesn_ordr
▶	3895	112	2.8755

Result:

We need a conversion rate (CVR) at least 4% for the campaign to make profitable. But we got 2.8% CVR. I decided to test by adjusting bids on the gsearch non-brand campaign to see how sensitive it is to bid changes.

- 3) I pulled the gsearch nonbrand trended session volume by week (date till 2012-05-10, this time, didn't bid on gsearch nonbrand)**

```
SELECT
    min(DATE(created_at)) as week_start_date,
    count(DISTINCT website_session_id) as session
FROM
    website_sessions
WHERE
    created_at < '2012-05-10' AND utm_source='gsearch' AND
    utm_campaign='nonbrand'
GROUP BY WEEK(created_at), YEAR(created_at);
```

OUTPUT:

week_start_date	session
2012-03-19	896
2012-03-25	956
2012-04-01	1152
2012-04-08	983
2012-04-15	621
2012-04-22	594
2012-04-29	681
2012-05-06	399

Result:

The above result gsearch nonbrand sensitive to the bid changes. We want to maximize the volume but don't want to spend more on ads.

- 4) Let's check by pulling the conversion rate from session to order by device type.

```
SELECT
    device_type,
    count(DISTINCT w.website_session_id) AS session,
    count(DISTINCT order_id) AS orders,
    count(DISTINCT order_id) / count(DISTINCT w.website_session_id) AS cnvrn_rt
FROM
    website_sessions AS w
    LEFT JOIN orders AS o ON w.website_session_id = o.website_session_id
WHERE
    w.created_at < '2012-05-11'
    AND utm_source = 'gsearch'
    AND utm_campaign = 'nonbrand'
GROUP BY
    device_type;
```

OUTPUT:

device_type	session	orders	cnvrn_rt
desktop	3911	146	0.0373
mobile	2492	24	0.0096

Result:

The conversion rate is higher on Desktop (3.73%) compared to Mobile (0.96%). Let's analyze the weekly trend for both device types.

- 5) I pulled weekly trends for desktop and mobile gsearch nonbrand (date: 2012-04-15 to 2012-06-09).

```
SELECT
    min(DATE(created_at)) AS week_start_date,
    count(CASE WHEN device_type='desktop' THEN website_session_id ELSE
    NULL END) AS dstop_session,
    count(CASE WHEN device_type='mobile' THEN website_session_id ELSE NULL
    END) AS mob_session
FROM
    website_sessions
WHERE
```

```

    created_at BETWEEN '2012-04-15' AND '2012-06-09' AND
utm_source='gsearch' AND utm_campaign='nonbrand'
GROUP BY
WEEK(created_at),YEAR(created_at);

```

OUTPUT:

week_start_date	dstop_session	mob_session
2012-04-15	383	238
2012-04-22	360	234
2012-04-29	425	256
2012-05-06	430	282
2012-05-13	403	214
2012-05-20	661	190
2012-05-27	585	183
2012-06-03	582	157

Result:

The bid applied on gsearch nonbrand desktop on 2012-05-10 is effective, and the number of website visitors has increased. This indicates that it is sensitive to bid changes

2) Analyzing the website performance:

1) Let's pull the top view pages (Date till: 2012-06-09)

```

select pageview_url, count(distinct website_session_id) as session
from website_pageviews
where
    created_at < '2012-06-09'
group by
    pageview_url
order by session desc;

```

OUTPUT:

pageview_url	session
/home	10403
/products	4239
/the-original-mr-fuzzy	3037
/cart	1306
/shipping	869
/billing	716
/thank-you-for-your-order	306

Result:

/Home, products and /the-original-mr-fuzzy are the top page visited by the customer.

- 2) List the top entry pages to our website

```
with
    top_entry_detail as (
        select
            website_session_id,
            min(website_pageview_id) as landeling_page
        from website_pageviews
        where
            created_at < '2012-06-12'
        group by
            website_session_id
    )

    select pageview_url, count(distinct t.website_session_id) as
    no_of_visitor
    from
        top_entry_detail as t
        left join website_pageviews as w on t.landeling_page =
    w.website_pageview_id
    group by
        pageview_url
    order by no_of_visitor desc;
```

OUTPUT:

	pageview_url	no_of_visitor
▶	/home	10714

Result:

All our traffic is landing first to home page. So let's find out is our customer purchasing the product.

3) Let's pull the bounce rate for traffic landing on the home page.

```
-- creating the table of first page entry of the visitor
-- counting the number of visitors entered to the first page
-- using order table count the number of orders

create temporary table top_entry
select
    website_session_id,
    min(website_pageview_id) as min_landing_page
from website_pageviews
where
    created_at < '2012-06-14'
group by
    website_session_id;

create temporary table no_of_visitors
select pageview_url, t.website_session_id
from
    top_entry as t
        left join website_pageviews as w on t.min_landing_page =
w.website_pageview_id;

create temporary table bounced_session
select n.website_session_id, n.pageview_url, count(w.website_pageview_id)
as count_of_page_view
from
    no_of_visitors as n
        left join website_pageviews as w on n.website_session_id =
w.website_session_id
group by
    n.website_session_id,
```

```

n.pageview_url
having
    count_of_page_view = 1;

select
    count(n.website_session_id) as session,
    count(count_of_page_view) as bounced_session,
    count(count_of_page_view) / count(n.website_session_id) as
percentage_bounced_session
from
    no_of_visitors as n
    left join bounced_session as b on n.website_session_id =
b.website_session_id;

```

OUTPUT:

	session	bounced_session	percentage_bounced_session
▶	11048	6538	0.5918

Result:

Bounce rate is 59.18% is high for paid search. So, let's introduce the new custom landing page.

- 4) **Based on the analysis, introduced new custom page (/lander-1) by 50/50 against the homepage for gsearch nonbrand. Let's pull the bounce rate for two groups.**

```

-- step1: for fair decision check the date of both page (/home and
/lander-1) should be same

-- step2: find first time visiting to the pages

-- step3: categories the no of visitors to the landing page

-- step4 find the bounced session who visited to the first page but not
to next page

-- step5: obtained landing_page, session, bounced session and
percentage of bounced session

```

```
select * from website_pageviews where pageview_url = '/lander-1';
```

OUTPUT:

website_pageview_id	created_at	website_session_id	pageview_url
23504	2012-06-19 00:35:54	11683	/lander-1
23506	2012-06-19 01:02:38	11685	/lander-1
23507	2012-06-19 01:31:57	11686	/lander-1

```
-- so, start date from 2012-06-19

create temporary table landing_page
select ws.website_session_id, min(website_pageview_id) as top_entry
from
    website_sessions as ws
    inner join website_pageviews as wp on ws.website_session_id =
wp.website_session_id
where
    ws.created_at between '2012-06-19' and '2012-07-28'
    and utm_source = "gsearch"
    and utm_campaign = "nonbrand"
group by
    website_session_id;

create temporary table top_entering_page
select l.website_session_id, pageview_url
from
    landing_page as l
    left join website_pageviews as w on l.top_entry =
w.website_pageview_id
where
    pageview_url in ('/lander-1', '/home');

create temporary table first_page_entry
select t.website_session_id, t.pageview_url,
count(w.website_pageview_id) as first_time_visited
from
    top_entering_page as t
    left join website_pageviews as w on t.website_session_id =
w.website_session_id
group by
    t.website_session_id,
    t.pageview_url
having
```

```

first_time_visited = 1;

select
    t.pageview_url,
    count(t.website_session_id) as session,
    count(f.first_time_visited) as bounced_session,
    count(first_time_visited) / count(t.website_session_id) as
percentage
from
    top_entering_page as t
    left join first_page_entry as f on t.website_session_id =
f.website_session_id
group by
    pageview_url
order by percentage desc;

```

OUTPUT:

pageview_url	session	bounced_session	percentage
/home	2261	1319	0.5834
/lander-1	2316	1233	0.5324

Result:

/home page has 58% bounce rate and /lander-1 has 53% bounce rate so there no much differences, but /lander-1 performing less bounce rate.

- 5) **Also, let's pull the volume of paid search nonbrand landing on /home and /lander-1 trended weekly. Also pull the overall paid search bounce rate trended by weekly.(/lander-1 started from 2012-06-19)**

```

create temporary table first_landing_page
select ws.website_session_id, min(website_pageview_id) as top_entry
from
    website_sessions as ws
    inner join website_pageviews as wp on ws.website_session_id =
wp.website_session_id
where
    utm_source = "gsearch"
    and utm_campaign = "nonbrand"
    and ws.created_at between "2012-06-01" and "2012-08-31"
group by
    ws.website_session_id;

```

```

create temporary table session_tab
select
    pageview_url as landing_page,
    f.website_session_id,
    created_at
from
    first_landing_page as f
    left join website_pageviews as w on top_entry =
w.website_pageview_id
where
    pageview_url in ("/home", "/lander-1");

create temporary table bounce_session
select s.website_session_id, s.landing_page,
count(w.website_pageview_id) as first_time_visited
from
    session_tab as s
    left join website_pageviews as w on s.website_session_id =
w.website_session_id
group by
    s.website_session_id,
    s.landing_page
having
    first_time_visited = 1;

select
    min(created_at) as start_date_weekly,
    count(first_time_visited) / count(s.landing_page) as
bounced_session,
    count(
        case
            when s.landing_page = "/home" then 1
            else null
        end
    ) as home_session,
    count(
        case
            when s.landing_page = "/lander-1" then 1
            else null
        end
    ) as lander_session
from
    session_tab as s
    left join bounce_session as b on s.website_session_id =
b.website_session_id

```

```

group by
week(created_at),
year(created_at);

```

OUTPUT:

start_date_weekly	bounced_session	home_session	lander_session
2012-06-01 00:05:11	0.6057	175	0
2012-06-03 00:43:23	0.5871	792	0
2012-06-10 00:00:22	0.6160	875	0
2012-06-17 00:34:49	0.5582	492	350
2012-06-24 00:27:52	0.5828	369	386
2012-07-01 00:10:01	0.5821	392	388
2012-07-08 00:47:28	0.5668	390	411
2012-07-15 00:09:07	0.5424	429	421
2012-07-22 00:39:03	0.5138	402	394
2012-07-29 00:58:12	0.4971	33	995
2012-08-05 00:11:26	0.5382	0	1087
2012-08-12 00:49:16	0.5140	0	998
2012-08-19 00:01:43	0.5010	0	1012
2012-08-26 00:30:14	0.5378	0	833

Result:

By looking at the bounce rate slightly reduced after introducing the lander-1

- 6) Let's build the full conversion funnel, how many customer visit make it each step. (start with /lander-1 page build the funnel all the way to thank you page for gsearch nonbrand)**

```

create temporary table lander_1_pages
select
    website_session_id,
    max(product_page) as product_made_it,
    max(mr_fuzzy_page) as fuzzy_made_it,
    max(cart_page) as cart_made_it,
    max(shipping_page) as shipping_made_it,
    max(billing_page) as billing_made_it,
    max(order_page) as order_made_it
from (
    select
        ws.website_session_id, (
            case
                when pageview_url = "/products" then 1

```

```

            else 0
        end
    ) as product_page, (
        case
            when pageview_url = "/the-original-mr-fuzzy" then 1
            else 0
        end
    ) as mr_fuzzy_page, (
        case
            when pageview_url = "/cart" then 1
            else 0
        end
    ) as cart_page, (
        case
            when pageview_url = "/shipping" then 1
            else 0
        end
    ) as shipping_page, (
        case
            when pageview_url = "/billing" then 1
            else 0
        end
    ) as billing_page, (
        case
            when pageview_url = "/thank-you-for-your-order"
then 1
            else 0
        end
    ) as order_page
from
    website_sessions as ws
    left join website_pageviews as wp on ws.website_session_id
= wp.website_session_id
    where
        utm_source = "gsearch"
        and utm_campaign = "nonbrand"
        and ws.created_at > "2012-08-05"
        and ws.created_at < "2012-09-05"
) as lander_1_page
group by
    website_session_id;

select
    count(website_session_id) as session,

```

```

count(distinct case when product_made_it=1 then website_session_id
else null end) as product_made_it,
count(distinct case when fuzzy_made_it=1 then website_session_id else
null end) as fuzzy_made_it,
count(distinct case when cart_made_it=1 then website_session_id else null
end)
    as cart_made_it,
count(distinct case when shipping_made_it=1 then website_session_id else
null end)
    as shipping_made_it,
count(distinct case when billing_made_it=1 then website_session_id else
null end)
    as billing_made_it,
count(distinct case when order_made_it=1 then website_session_id else null
end)
    as order_made_it,
count(distinct case when product_made_it=1 then website_session_id else
null end)/count(website_session_id) as product_made_it_per,
    count(distinct case when fuzzy_made_it=1 then website_session_id else
null end) /count(distinct case when product_made_it=1 then
website_session_id else null end)
    as fuzzy_made_it_per,
count(distinct case when cart_made_it=1 then website_session_id else null
end)/count(distinct case when fuzzy_made_it=1 then website_session_id else
null end)
    as cart_made_it_per,
count(distinct case when shipping_made_it=1 then website_session_id else
null end)/count(distinct case when cart_made_it=1 then website_session_id
else null end)
    as shipping_made_it,
count(distinct case when billing_made_it=1 then website_session_id else
null end)/count(distinct case when shipping_made_it=1 then
website_session_id else null end)
    as billing_made_it_per,
count(distinct case when order_made_it=1 then website_session_id else null
end)/count(distinct case when billing_made_it=1 then website_session_id
else null end)
    as order_made_it
from lander_1_pages;

```

OUTPUT:

	session	product_made_it	fuzzy_made_it	cart_made_it	shipping_made_it	billing_made_it	order_made_it
▶	4493	2115	1567	683	455	361	158

product_made_it_per	fuzzy_made_it_per	cart_made_it_per	shipping_made_it	billing_made_it_per	order_made_it
0.4707	0.7409	0.4359	0.6662	0.7934	0.4377

Result:

By looking at the percentage value, conversion from fuzzy page to cart page has less conversion and from billing to order has less conversion. So, let's make new custom billing page (/billing-2)

- 7) Updated the billing page based on previous funnel analysis. Let's check weather billing-2 is doing better than billing-1**

```
-- let me check from where the /billing-2 webpage introduced
```

```
select * from website_pageviews where pageview_url = "/billing-2";
```

OUTPUT:

website_pageview_id	created_at	website_session_id	pageview_url
53550	2012-09-10 00:13:05	25325	/billing-2
53628	2012-09-10 08:36:13	25358	/billing-2
53753	2012-09-10 12:22:36	25411	/billing-2

```
-- its started from "2012-09-10"
```

```
-- step 1: filter the data contain gsearch non brand and filter the date range where both billing and billing-2 started
```

```
-- step 2: obtain the session_id where the customer ordered and group by the page view_url
```

```
-- step3: count the number of sessions, customer visiting the billing page and billing to order
```

```
select  
    pageview_url as billing_session,
```

```

count(
    billing_data.website_session_id
) as session,
count(order_id) as orders,
count(order_id) / count(
    billing_data.website_session_id
) as billing_to_order
from (
    select ws.website_session_id, pageview_url
    from
        website_sessions as ws
        left join website_pageviews as wp on ws.website_session_id =
wp.website_session_id
    where
        wp.created_at > '2012-09-10'
        and ws.created_at < '2012-11-10'
        and utm_source = 'gsearch'
        and utm_campaign = 'nonbrand'
        and pageview_url in ("/billing", "/billing-2")
) as billing_data
left join orders on billing_data.website_session_id =
orders.website_session_id
group by
pageview_url

```

OUTPUT:

billing_session	session	orders	billing_to_order
/billing-2	396	242	0.6111
/billing	401	176	0.4389

Result:

So /billing-2 has high converting rate compared to the /billing. This helps to reduce the bounce rate and try to improve the billing to order conversion rate.

Conclusion:

This project gave me a clear understanding of how users interact with the website and what drives sales. The insights helped the business make smarter marketing decisions, which improved engagement and increased conversions