ADVANCED SQL ANALYSIS

PROJECT

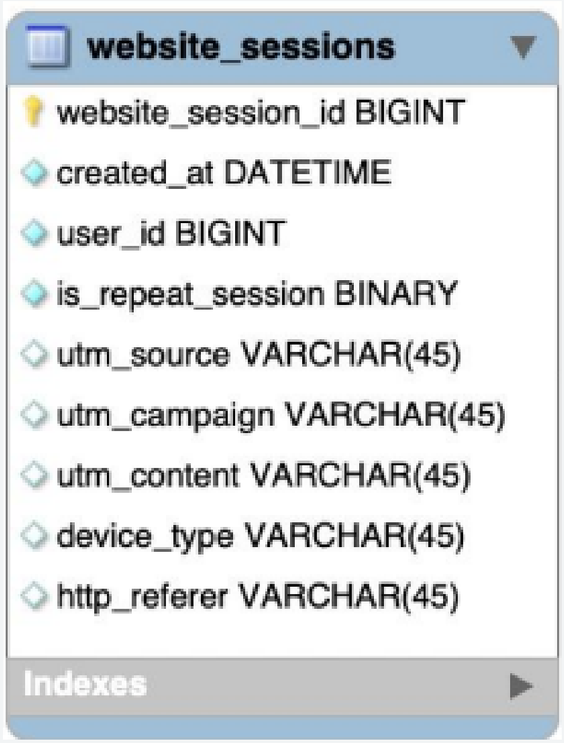
WWW.LINKEDIN.COM/IN/SAURAVSINGHHHH/

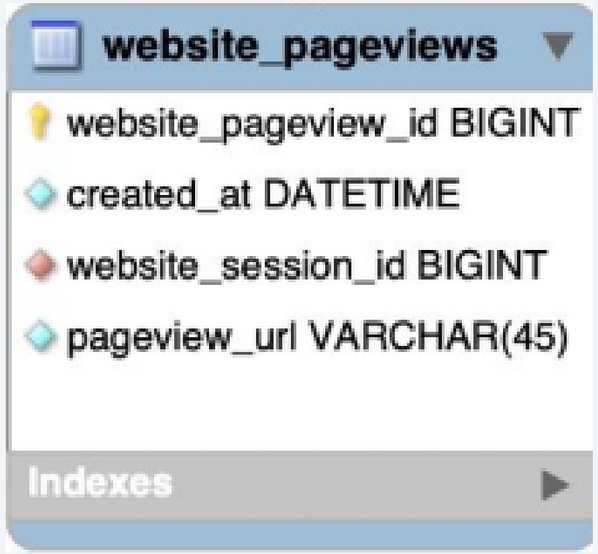
OBJECTIVE:

ANALYZING
TRAFFIC SOURCES &
WEBSITE PERFORMANCE

@SAURAVSINGHHHH

UNDERSTAND
MAVEN FUZZY FACTORY DATABASE





In this project, we will use these tables

@SAURAVSINGHHHH

TOPICS COVERED IN THIS PROJECT

Sr no.	Used	Sr no.	Used
1	*Finding Top Traffic Sources*	5	*Traffic Source Segment Trending*
2	*Traffic Conversion Rates*	6	*Identifying Top Website Pages*
3	*Traffic Source Trending*	7	*Building Conversion Funnels*
4	*Traffic Source Bid Optimization*	8	*Analyzing Conversion Funnel Tests*

@SAURAVSINGHHHH

QUESTIONS TO BE ANSWERED IN THIS ANALYSIS

[1/8]

Gsearch seems to be the biggest driver of our business. Could you pull monthly trends for gsearch sessions and orders so that we can showcase the growth there?

@SAURAVSINGHHHH

trends for gsearch sessions and orders

```sql
SELECT
    YEAR(website_sessions.created_at) AS yr,
    MONTH(website_sessions.created_at) AS mo,
    COUNT(DISTINCT website_sessions.website_session_id) AS sessions,
    COUNT(DISTINCT orders.order_id) AS orders,
    COUNT(DISTINCT orders.order_id)/COUNT(DISTINCT website_sessions.website_session_id) AS conv_rate
FROM website_sessions
    LEFT JOIN orders
        ON orders.website_session_id = website_sessions.website_session_id
WHERE website_sessions.created_at < '2012-11-27'
    AND website_sessions.utm_source = 'gsearch'
GROUP BY 1,2;
```

yr	mo	sessions	orders	conv_rate
2012	3	1860	60	0.0323
2012	4	3574	92	0.0257
2012	5	3410	97	0.0284
2012	6	3578	121	0.0338
2012	7	3811	145	0.0380
2012	8	4877	184	0.0377
2012	9	4491	188	0.0419
2012	10	5534	234	0.0423
2012	11	8889	373	0.0420

Here we got year, month and sessions volumes are growing by month and orders with conversion rates is growing as well substantially. This is nice improvement for a business.

@SAURAVSINGHHHH

QUESTIONS TO BE ANSWERED IN THIS ANALYSIS

[2/8]

Next, it would be great to see a similar monthly trend for Gsearch, but this time splitting out nonbrand and brand campaigns separately. I am wondering if brand is picking up at all. If so, this is a good story to tell.

@SAURAVSINGHHHH

splitting out nonbrand and brand campaigns separately

```sql
SELECT
    YEAR(website_sessions.created_at) AS yr,
    MONTH(website_sessions.created_at) AS mo,
    COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' THEN website_sessions.website_session_id
    ELSE NULL END) AS nonbrand_sessions,
    COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' THEN orders.order_id
    ELSE NULL END) AS nonbrand_orders,
    COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN website_sessions.website_session_id
    ELSE NULL END) AS brand_sessions,
    COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN orders.order_id
    ELSE NULL END) AS brand_orders
FROM website_sessions
    LEFT JOIN orders
        ON orders.website_session_id = website_sessions.website_session_id
WHERE website_sessions.created_at < '2012-11-27'
    AND website_sessions.utm_source = 'gsearch'
GROUP BY 1,2;
```

yr	mo	nonbrand_sessions	nonbrand_orders	brand_sessions	brand_orders
2012	3	1852	60	8	0
2012	4	3509	86	65	6
2012	5	3295	91	115	6
2012	6	3439	114	139	7
2012	7	3660	136	151	9
2012	8	4673	174	204	10
2012	9	4227	172	264	16
2012	10	5197	219	337	15
2012	11	8506	356	383	17

Brand campaign is something like someone going to search engine and searching for our business. But in this analysis it can be seen that most sessions are from nonbrand campaign.

@SAURAVSINGHHHH

QUESTIONS TO BE ANSWERED IN THIS ANALYSIS

[3/8]

While we’re on Gsearch, could you dive into nonbrand, and pull monthly sessions and orders split by device type? I want to flex our analytical muscles a little and show the board we really know our traffic sources.

@SAURAVSINGHHHH

monthly sessions and orders split by device type

```sql
SELECT
    YEAR(website_sessions.created_at) AS yr,
    MONTH(website_sessions.created_at) AS mo,
    COUNT(DISTINCT CASE WHEN device_type = 'desktop' THEN website_sessions.website_session_id
    ELSE NULL END) AS desktop_sessions,
    COUNT(DISTINCT CASE WHEN device_type = 'desktop' THEN orders.order_id
    ELSE NULL END) AS desktop_orders,
    COUNT(DISTINCT CASE WHEN device_type = 'mobile' THEN website_sessions.website_session_id
    ELSE NULL END) AS mobile_sessions,
    COUNT(DISTINCT CASE WHEN device_type = 'mobile' THEN orders.order_id
    ELSE NULL END) AS mobile_orders
FROM website_sessions
    LEFT JOIN orders
        ON orders.website_session_id = website_sessions.website_session_id
WHERE website_sessions.created_at < '2012-11-27'
    AND website_sessions.utm_source = 'gsearch'
    AND website_sessions.utm_campaign = 'nonbrand'
GROUP BY 1,2;
```

yr	mo	desktop_sessions	desktop_orders	mobile_sessions	mobile_orders
2012	3	1128	50	724	10
2012	4	2139	75	1370	11
2012	5	2276	83	1019	8
2012	6	2673	106	766	8
2012	7	2774	122	886	14
2012	8	3515	165	1158	9
2012	9	3171	155	1056	17
2012	10	3934	201	1263	18
2012	11	6457	323	2049	33

Here we can see a lot more desktop sessions from beginning as compared to mobile sessions. Initially we can see desktop sessions and mobile sessions were in 2:1 but after a couple months it increased to more than 3:1 ratios. Similarly , in orders desktop to mobile orders in beginning was 5:1 ratios and it increased to 10:1 ratios.

@SAURAVSINGHHHH

QUESTIONS TO BE ANSWERED IN THIS ANALYSIS

[4/8]

I'm worried that one of our more pessimistic board members may be concerned about the large % of traffic from Gsearch. Can you pull monthly trends for Gsearch, alongside monthly trends for each of our other channels?

I am concerned about large % of traffic from Gsearch

@SAURAVSINGHHHH

to see the traffic we're getting

```sql
SELECT DISTINCT
    utm_source,
    utm_campaign,
    http_referer
FROM website_sessions
WHERE website_sessions.created_at < '2012-11-27';
```

utm_source	utm_campaign	http_referer
gsearch	nonbrand	https://www.gsearch.com
NULL	NULL	NULL
gsearch	brand	https://www.gsearch.com
NULL	NULL	https://www.gsearch.com
bsearch	brand	https://www.bsearch.com
NULL	NULL	https://www.bsearch.com
bsearch	nonbrand	https://www.bsearch.com

Part 1 : finding the various utm sources and referers to see the traffic we're getting

When utm_source, utm_campaign and http_referer is null then there is direct type in traffic. When only utm_source , utm_campaign are null then it shows it is organic search traffic.

@SAURAVSINGHHHH

Various sessions comparison

```sql
SELECT
    YEAR(website_sessions.created_at) AS yr,
    MONTH(website_sessions.created_at) AS mo,
    COUNT(DISTINCT CASE WHEN utm_source = 'gsearch' THEN website_sessions.website_session_id
    ELSE NULL END) AS gsearch_paid_sessions,
    COUNT(DISTINCT CASE WHEN utm_source = 'bsearch' THEN website_sessions.website_session_id
    ELSE NULL END) AS bsearch_paid_sessions,
    COUNT(DISTINCT CASE WHEN utm_source IS NULL AND http_referer IS NOT NULL
    THEN website_sessions.website_session_id ELSE NULL END) AS organic_search_sessions,
    COUNT(DISTINCT CASE WHEN utm_source IS NULL AND http_referer IS NULL
    THEN website_sessions.website_session_id ELSE NULL END) AS direct_type_in_sessions
FROM website_sessions
    LEFT JOIN orders
        ON orders.website_session_id = website_sessions.website_session_id
WHERE website_sessions.created_at < '2012-11-27'
GROUP BY 1,2;
```

yr	mo	gsearch_paid_sessions	bsearch_paid_sessions	organic_search_sessions	direct_type_in_sessions
2012	3	1860	2	8	9
2012	4	3574	11	78	71
2012	5	3410	25	150	151
2012	6	3578	25	190	170
2012	7	3811	44	207	187
2012	8	4877	705	265	250
2012	9	4491	1439	331	285
2012	10	5534	1781	428	440
2012	11	8889	2840	536	485

Part 2 :
sessions comparison

Here we can see gsearch_paid_sessions are building over time, bsearch_paid_sessions is much more increased then it was before. Orgainic search and direct type in session are growing more and it show engagement of customers to our website without any paid marketing.

@SAURAVSINGHHHH

QUESTIONS TO BE ANSWERED IN THIS ANALYSIS

[5/8]

I'd like to tell the story of our website performance improvements over the course of the first 8 months. Could you pull session to order conversion rates, by month?

@SAURAVSINGHHHH

session to order conversion rates, by month

```sql
SELECT
    YEAR(website_sessions.created_at) AS yr,
    MONTH(website_sessions.created_at) AS mo,
    COUNT(DISTINCT website_sessions.website_session_id) AS sessions,
    COUNT(DISTINCT orders.order_id) AS orders,
    COUNT(DISTINCT orders.order_id)/COUNT(DISTINCT website_sessions.website_session_id) AS conversion_rate
FROM website_sessions
    LEFT JOIN orders
        ON orders.website_session_id = website_sessions.website_session_id
WHERE website_sessions.created_at < '2012-11-27'
GROUP BY 1,2;
```

yr	mo	sessions	orders	conversion_rate
2012	3	1879	60	0.0319
2012	4	3734	99	0.0265
2012	5	3736	108	0.0289
2012	6	3963	140	0.0353
2012	7	4249	169	0.0398
2012	8	6097	228	0.0374
2012	9	6546	287	0.0438
2012	10	8183	371	0.0453
2012	11	12750	561	0.0440

Here it can be seen that sessions and orders are increased over a month and conversion rate increased from 3% to 4% over a month.

@SAURAVSINGHHHH

QUESTIONS TO BE ANSWERED IN THIS ANALYSIS

[6/8]

For the gsearch lander test, please estimate the revenue that test earned us
(Hint:
Look at the increase in CVR from the test (Jun 19 – Jul 28), and use nonbrand sessions and revenue since then to calculate incremental value)

@SAURAVSINGHHHH

```sql
SELECT
    MIN(website_pageview_id) AS first_test_pv
FROM website_pageviews
WHERE pageview_url = '/lander-1';
```

first_test_pv
23504

PART:[1/7]

we'll get the first pageview id

```sql
CREATE TEMPORARY TABLE first_test_pageviews
SELECT
  website_pageviews.website_session_id,
    MIN(website_pageviews.website_pageview_id) AS min_pageview_id
FROM website_pageviews
  INNER JOIN website_sessions
    ON website_sessions.website_session_id = website_pageviews.website_session_id
    AND website_sessions.created_at < '2012-07-28' -- prescribed by the assignment
    AND website_pageviews.website_pageview_id ⩾ 23504 -- first page_view
        AND utm_source = 'gsearch'
        AND utm_campaign = 'nonbrand'
GROUP BY
  website_pageviews.website_session_id;
```

website_session_id	min_pageview_id
11683	23504
11684	23505
11685	23506

PART:[2/7]

Here we will create temporary table as first_test_pageviews. It have website session id and min pageview id

@SAURAVSINGHHHH

```sql
CREATE TEMPORARY TABLE nonbrand_test_sessions_w_landing_pages
SELECT
    first_test_pageviews.website_session_id,
    website_pageviews.pageview_url AS landing_page
FROM first_test_pageviews
    LEFT JOIN website_pageviews
        ON website_pageviews.website_pageview_id = first_test_pageviews.min_pageview_id
WHERE website_pageviews.pageview_url IN ('/home','/lander-1');
```

website_session_id	landing_page
11683	/lander-1
11684	/home
11685	/lander-1
11686	/lander-1
11687	/home
11688	/home
11689	/lander-1
11690	/home
11691	/lander-1

PART:[3/7]

we'll bring in the landing page to each session, but restricting to home or lander-1 this time.
This will be used to figure out which pages someone saw and it will help to understand which page perform better.

@SAURAVSINGHHHH

```sql
CREATE TEMPORARY TABLE nonbrand_test_sessions_w_orders
SELECT
    nonbrand_test_sessions_w_landing_pages.website_session_id,
    nonbrand_test_sessions_w_landing_pages.landing_page,
    orders.order_id AS order_id

FROM nonbrand_test_sessions_w_landing_pages
LEFT JOIN orders
ON orders.website_session_id = nonbrand_test_sessions_w_landing_pages.website_session_id;
```

website_session_id	landing_page	order_id
11683	/lander-1	NULL
11684	/home	NULL
11685	/lander-1	NULL
11686	/lander-1	NULL
11687	/home	NULL

PART:[4/7]

we made a temporary table to bring in orders. This will be used to find final performance counts.

@SAURAVSINGHHHH

```sql
SELECT
    landing_page,
    COUNT(DISTINCT website_session_id) AS sessions,
    COUNT(DISTINCT order_id) AS orders,
    COUNT(DISTINCT order_id)/COUNT(DISTINCT website_session_id) AS conv_rate
FROM nonbrand_test_sessions_w_orders
GROUP BY 1;
```

landing_page	sessions	orders	conv_rate
/home	2261	72	0.0318
/lander-1	2316	94	0.0406

PART:[5/7]

Here, we find the difference between conversion rates .
0.0319 for /home, vs .0406 for /lander-1
0.0087 additional orders per session
It shows that we can more focus on lander-1 page

@SAURAVSINGHHHH

```sql
SELECT
    MAX(website_sessions.website_session_id) AS most_recent_gsearch_nonbrand_home_pageview
FROM website_sessions
    LEFT JOIN website_pageviews
        ON website_pageviews.website_session_id = website_sessions.website_session_id
WHERE utm_source = 'gsearch'
    AND utm_campaign = 'nonbrand'
    AND pageview_url = '/home'
    AND website_sessions.created_at < '2012-11-27';
```

most_recent_gsearch_nonbrand_home_pageview
17145

PART:[6/7]

Here, we got the most recent pageview for gsearch nonbrand where the traffic was sent to /home
max website_session_id = 17145
That is the maximum session id of gsearch nonbrand campaign going to /home page. Since then all the traffic is rerouted elsewhere.

@SAURAVSINGHHHH

```sql
SELECT
    COUNT(website_session_id) AS sessions_since_test
FROM website_sessions
WHERE created_at < '2012-11-27'
    AND website_session_id > 17145 -- last /home session
    AND utm_source = 'gsearch'
    AND utm_campaign = 'nonbrand';
```

PART:[7/7]

sessions_since_test
22972

22,972 website sessions since the test

X .0087 incremental conversion = 202 incremental orders since July 29
roughly 4 months, so roughly 50 extra orders per month. Not bad!

Hopefully this analysis make sense, we have improved the performance of website by changing over to the new page. We have quantified that improvements by 0.0087 amount of lift of orders generated per session.

Found total number of session 22,972 * 0.0087 we get 202 incremental orders.
If we talk about orders per month , it generates 50 extra orders per month.

@SAURAVSINGHHHH

QUESTIONS TO BE ANSWERED IN THIS ANALYSIS

[7/8]

For the landing page test you analyzed previously, it would be great to show a full conversion funnel from each of the two pages to orders. You can use the same time period you analyzed last time (Jun 19 – Jul 28).

@SAURAVSINGHHHH

PART:[1/3]

```sql
CREATE TEMPORARY TABLE session_level_made_it_flagged
SELECT
  website_session_id,
    MAX(homepage) AS saw_homepage,
    MAX(custom_lander) AS saw_custom_lander,
    MAX(products_page) AS product_made_it,
    MAX(mrfuzzy_page) AS mrfuzzy_made_it,
    MAX(cart_page) AS cart_made_it,
    MAX(shipping_page) AS shipping_made_it,
    MAX(billing_page) AS billing_made_it,
    MAX(thankyou_page) AS thankyou_made_it
FROM(
SELECT
  website_sessions.website_session_id,
    website_pageviews.pageview_url,
    -- website_pageviews.created_at AS pageview_created_at,
    CASE WHEN pageview_url = '/home' THEN 1 ELSE 0 END AS homepage,
    CASE WHEN pageview_url = '/lander-1' THEN 1 ELSE 0 END AS custom_lander,
    CASE WHEN pageview_url = '/products' THEN 1 ELSE 0 END AS products_page,
    CASE WHEN pageview_url = '/the-original-mr-fuzzy' THEN 1 ELSE 0 END AS mrfuzzy_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 thankyou_page
FROM website_sessions
  LEFT JOIN website_pageviews
    ON website_sessions.website_session_id = website_pageviews.website_session_id
WHERE website_sessions.utm_source = 'gsearch'
  AND website_sessions.utm_campaign = 'nonbrand'
    AND website_sessions.created_at < '2012-07-28'
    AND website_sessions.created_at > '2012-06-19'
ORDER BY
  website_sessions.website_session_id,
    website_pageviews.created_at
) AS pageview_level

GROUP BY
  website_session_id;
SELECT * FROM session_level_made_it_flagged;
```

website_session_id	saw_homepage	saw_custom_lander	product_made_it	mrfuzzy_made_it	cart_made_it	shipping_made_it	billing_made_it	thankyou_made_it
11683	0	1	0	0	0	0	0	0
11684	1	0	0	0	0	0	0	0
11685	0	1	0	0	0	0	0	0
11686	0	1	1	0	0	0	0	0
11687	1	0	0	0	0	0	0	0
11688	1	0	0	0	0	0	0	0
11689	0	1	1	1	0	0	0	0
11690	1	0	0	0	0	0	0	0
11691	0	1	1	0	0	0	0	0
11692	0	1	0	0	0	0	0	0

Here, we created temporary table to find which session completed upto which page. In this we added flags as 0 and 1 to find which page it is.

@SAURAVSINGHHHH

```sql
SELECT
    CASE
        WHEN saw_homepage = 1 THEN 'saw_homepage'
        WHEN saw_custom_lander = 1 THEN 'saw_custom_lander'
        ELSE 'uh oh... check logic'
    END AS segment,
    COUNT(DISTINCT website_session_id) AS sessions,
    COUNT(DISTINCT CASE WHEN product_made_it = 1 THEN website_session_id ELSE NULL END) AS to_products,
    COUNT(DISTINCT CASE WHEN mrfuzzy_made_it = 1 THEN website_session_id ELSE NULL END) AS to_mrfuzzy,
    COUNT(DISTINCT CASE WHEN cart_made_it = 1 THEN website_session_id ELSE NULL END) AS to_cart,
    COUNT(DISTINCT CASE WHEN shipping_made_it = 1 THEN website_session_id ELSE NULL END) AS to_shipping,
    COUNT(DISTINCT CASE WHEN billing_made_it = 1 THEN website_session_id ELSE NULL END) AS to_billing,
    COUNT(DISTINCT CASE WHEN thankyou_made_it = 1 THEN website_session_id ELSE NULL END) AS to_thankyou
FROM session_level_made_it_flagged
GROUP BY 1;
```

segment	sessions	to_products	to_mrfuzzy	to_cart	to_shipping	to_billing	to_thankyou
saw_custom_lander	2316	1083	772	348	231	197	94
saw_homepage	2261	942	684	296	200	168	72

PART:[2/3]

Here we can see that more orders were placed through lander page as compared to homepage.

@SAURAVSINGHHHH

```sql
SELECT
    CASE
        WHEN saw_homepage = 1 THEN 'saw_homepage'
        WHEN saw_custom_lander = 1 THEN 'saw_custom_lander'
        ELSE 'uh oh... check logic'
    END AS segment,
    COUNT(DISTINCT CASE WHEN product_made_it = 1 THEN website_session_id ELSE NULL END)
        /COUNT(DISTINCT website_session_id) AS lander_click_rt,
    COUNT(DISTINCT CASE WHEN mrfuzzy_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 products_click_rt,
    COUNT(DISTINCT CASE WHEN cart_made_it = 1 THEN website_session_id ELSE NULL END)
        /COUNT(DISTINCT CASE WHEN mrfuzzy_made_it = 1 THEN website_session_id ELSE NULL END) AS mrfuzzy_click_rt,
    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 cart_click_rt,
    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 shipping_click_rt,
    COUNT(DISTINCT CASE WHEN thankyou_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 billing_click_rt
FROM session_level_made_it_flagged
GROUP BY 1;
```

PART:[3/3]

segment	lander_click_rt	products_click_rt	mrfuzzy_click_rt	cart_click_rt	shipping_click_rt	billing_click_rt
saw_custom_lander	0.4676	0.7128	0.4508	0.6638	0.8528	0.4772
saw_homepage	0.4166	0.7261	0.4327	0.6757	0.8400	0.4286

Here it shows clickthrough rates of one page to another. It is a conversion funnel of moving from initial page to final billing page.

@SAURAVSINGHHHH

QUESTIONS TO BE ANSWERED IN THIS ANALYSIS

[8/8]

I’d love for you to quantify the impact of our billing test, as well. Please analyze the lift generated from the test (Sep 10 – Nov 10), in terms of revenue per billing page session, and then pull the number of billing page sessions for the past month to understand monthly impact.

@SAURAVSINGHHHH

```sql
SELECT
    website_pageviews.website_session_id,
    website_pageviews.pageview_url AS billing_version_seen,
    orders.order_id,
    orders.price_usd
FROM website_pageviews
    LEFT JOIN orders
        ON orders.website_session_id = website_pageviews.website_session_id
WHERE website_pageviews.created_at > '2012-09-10' -- prescribed in assignment
    AND website_pageviews.created_at < '2012-11-10' -- prescribed in assignment
    AND website_pageviews.pageview_url IN ('/billing','/billing-2');
```

PART:[1/3]

website_session_id	billing_version_seen	order_id	price_usd
25325	/billing-2	871	49.99
25343	/billing	872	49.99
25353	/billing	873	49.99
25358	/billing-2	874	49.99
25368	/billing	875	49.99
25393	/billing	876	49.99
25411	/billing-2	877	49.99
25454	/billing-2	878	49.99
25459	/billing-2	NULL	NULL
25468	/billing	NULL	NULL

Here, we can see billing versions with order id and price. you have noticed that some billing versions have null order id and price. Reason behind it is the purchase of product haven't proceeded.
We can connect our customer supports to that user.

@SAURAVSINGHHHH

PART:[2/3]

```sql
SELECT
    billing_version_seen,
    COUNT(DISTINCT website_session_id) AS sessions,
    SUM(price_usd)/COUNT(DISTINCT website_session_id) AS revenue_per_billing_page_seen
 FROM(
SELECT
    website_pageviews.website_session_id,
    website_pageviews.pageview_url AS billing_version_seen,
    orders.order_id,
    orders.price_usd
FROM website_pageviews
    LEFT JOIN orders
        ON orders.website_session_id = website_pageviews.website_session_id
WHERE website_pageviews.created_at > '2012-09-10' -- prescribed in assignment
    AND website_pageviews.created_at < '2012-11-10' -- prescribed in assignment
    AND website_pageviews.pageview_url IN ('/billing','/billing-2')
) AS billing_pageviews_and_order_data
GROUP BY 1;
```

billing_version_seen	sessions	revenue_per_billing_page_seen
/billing	657	22.826484
/billing-2	654	31.339297

We have Analyzed that
$22.83 revenue per billing page seen for the old version

$31.34 for the new version

LIFT: $8.51 per billing page view

We can now move our overall website traffic towards billing-2 page as it is has more user friendly payment system which increases revenue per billing page

@SAURAVSINGHHHH

```sql
SELECT
  COUNT(website_session_id) AS billing_sessions_past_month
FROM website_pageviews
WHERE website_pageviews.pageview_url IN ('/billing','/billing-2')
  AND created_at BETWEEN '2012-10-27' AND '2012-11-27' -- past month
;
```

PART:[3/3]

billing_sessions_past_month
1193

Here we can see 1,193 billing sessions past month

In previous analysis we seen a LIFT: $8.51 per billing session

Total billing over past month = billing session * LIFT
= 1,193 * $8.51
= $10,152.43
VALUE OF BILLING TEST: $10,152.43 over the past month

Now, we can conclude that modification of billing page helped alot to generate more revenue than previous billing page.

@SAURAVSINGHHHH

THANK’S FOR REVIEWING

Connect with me on:

WWW.LINKEDIN.COM/IN/SAURAVSINGHHHH/

GITHUB.COM/SAURAV2021C

TWITTER.COM/SAURAVSINGHHHHS