

# "Crunching Numbers for Maven Fuzzy Factory, an eCommerce Excellence: A Data-Driven Analysis of Sales and Trends"

## Introduction.

As an Analyst of the startup team, how to work with the CEO, the Head of Marketing, and the Website Manager to help steer the business. I will analyze and optimize marketing channels, measure and test website conversion performance, and use data to understand the impact of new product launches.

Maven Fuzzy Factory has been live for ~8 months, and CEO is due to present company performance metrics to the board further week. I'll be the one tasked with preparing relevant metrics to show the company's promising growth.

## Objective.

- Tell the story of your company's growth, using trended performance data.
- Use the database to explain some of the details around your growth story and quantify the revenue impact of some of your wins.
- Analyze current performance and use that data available to assess upcoming opportunities.

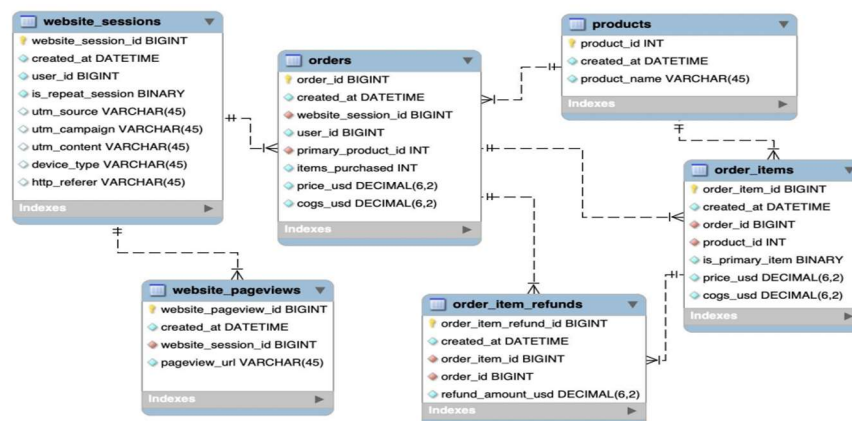
## Stakeholder's Expectations.

CEO needs some help preparing a presentation for the board meeting next week. The board would like to have a better understanding of our growth story over our first 8 months. This will also be a good excuse to show off our analytical capabilities a bit.

## Data description.

We will be working with six related tables, which contain eCommerce data about:

- Website Activity
- Products
- Orders and Refunds



## Methodology.

- SQL: Extract and analyze website traffic and performance data from the Maven Fuzzy Factory database to quantify the company's growth, and to tell the story of how you have been able to generate that growth.

**Exploratory Data Analysis (Stakeholder's requirements)** : For Analysis purposes, I would like to answer the following questions in order to get some insights from the data, and with some basic analytical interpretations I am going to help stakeholders make some decisions about the business.

- 1) Gsearch seems to be the biggest driver of our business. Pull monthly trends for gsearch sessions and orders to showcase the growth there.

```
SELECT
    MONTH(ws.created_at),
    COUNT(DISTINCT ws.website_session_id) AS sessions,
    COUNT(DISTINCT os.order_id) AS orders
FROM
    website_sessions ws
LEFT JOIN
    orders os
ON
    ws.website_session_id = os.website_session_id
WHERE
    ws.utm_source = 'gsearch' AND
    ws.created_at < '2012-11-27' -- According to date of request.
GROUP BY 1
```

	MONTH(ws.created_at)	sessions	orders
▶	3	1860	60
	4	3574	92
	5	3410	97
	6	3578	121
	7	3811	145
	8	4877	184
	9	4491	188
	10	5534	234
	11	8889	373

- 2) Next, it would be great to see a similar monthly trend for Gsearch, but this time splitting out nonbrand and brand campaigns separately. To have a better idea if brand is picking up at all. If so, this is a good story to tell.

```
SELECT
  ws.utm_campaign,
  MONTH(ws.created_at),
  COUNT(DISTINCT ws.website_session_id) AS sessions,
  COUNT(DISTINCT os.order_id) AS orders
FROM
  website_sessions ws
LEFT JOIN
  orders os
ON
  ws.website_session_id = os.website_session_id
WHERE
  ws.utm_source = 'gsearch' AND
  ws.created_at < '2012-11-27' -- According to the date requested.
GROUP BY 1,2
```

utm_campaign	MONTH(ws.created_at)	sessions	orders
brand	3	8	0
brand	4	65	6
brand	5	115	6
brand	6	139	7
brand	7	151	9
brand	8	204	10
brand	9	264	16
brand	10	337	15
brand	11	383	17
nonbrand	3	1852	60
nonbrand	4	3509	86
nonbrand	5	3295	91
nonbrand	6	3439	114
nonbrand	7	3660	136
nonbrand	8	4673	174
nonbrand	9	4227	172
nonbrand	10	5197	219
nonbrand	11	8506	356

3) While we're on Gsearch, Let's dive into nonbrand, and pull monthly sessions and orders split by device 3 type? To flex our analytical muscles a little and show the board we really know the traffic source.

```
SELECT
    MONTH(ws.created_at),
    COUNT(DISTINCT CASE WHEN device_type = 'desktop' THEN ws.website_session_id ELSE NULL END) AS
desktop_sessions,
    COUNT(DISTINCT CASE WHEN device_type = 'desktop' THEN os.website_session_id ELSE NULL END) AS
desktop_orders,
    COUNT(DISTINCT CASE WHEN device_type = 'mobile' THEN ws.website_session_id ELSE NULL END) AS
mobile_sessions,
    COUNT(DISTINCT CASE WHEN device_type = 'mobile' THEN os.website_session_id ELSE NULL END) AS
mobile_orders
FROM
    website_sessions ws
LEFT JOIN
    orders os
ON
    ws.website_session_id = os.website_session_id
WHERE
    ws.utm_source = 'gsearch' AND
    ws.created_at < '2012-11-27' AND -- According to date requested
    ws.utm_campaign = 'nonbrand'
GROUP BY 1
```

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

#### 4) Landing Page Analysis:

Landing page analysis and testing is about understanding the performance of your key landing pages and then testing to improve your results.

We have first landing page, “/home” and later launched, “/lander-1”

For the gsearch lander test, Let's estimate the revenue that test (launching “lander\_1) earned us. By looking at the increase in conversion rates from the test (Jun 19 – Jul 28), and use nonbrand sessions and revenue since then we can calculate incremental value.

```
-- Finding first pageview where lander 1 was seen.
SELECT
  MIN(website_pageview_id)
FROM
  website_pageviews
WHERE pageview_url = '/lander-1'
```

First **pageview-id** for lander 1 is **23504**.

now, creating temp table for min page view id for each session.

```
CREATE TEMPORARY TABLE Entry_page_view_1
SELECT
  wp.website_session_id,
  MIN(wp.website_pageview_id) AS First_pageview_id
FROM
  website_pageviews wp
INNER JOIN
  website_sessions ws
ON
  wp.website_session_id = ws.website_session_id
WHERE
  wp.created_at < '2012-07-28' AND -- according to the date requested
  wp.website_pageview_id >= '23504' AND -- First pageview-id for lander-1
  ws.utm_source = 'gsearch' AND
  ws.utm_campaign = 'nonbrand'
GROUP BY 1
```

Now, finding urls for each pageview-id and only subtracting ones which has either home or lander-1 urls.

```
CREATE TEMPORARY TABLE pageview_id_url
SELECT
  ep.website_session_id,
  ep.First_pageview_id,
  wp.pageview_url
FROM
  Entry_page_view_1 ep
LEFT JOIN
  website_pageviews wp
ON
  ep.First_pageview_id = wp.website_pageview_id
WHERE
  wp.pageview_url IN ('/home' , '/lander-1')
```

Now, using subquery to attach order id for respective website sessions from pageview\_id\_url table, then counting sessions, orders for each landing page.

```
SELECT
    pageview_url,
    COUNT(DISTINCT website_session_id) AS total_sessions,
    COUNT(DISTINCT order_id) AS total_orders,
    (COUNT(DISTINCT order_id)/COUNT(DISTINCT website_session_id))*100 AS CVR
FROM (
    SELECT
        pu.website_session_id,
        pu.pageview_url,
        os.order_id
    FROM
        pageview_id_url pu
    LEFT JOIN
        orders os
    ON
        pu.website_session_id = os.website_session_id AS landing_page_orders
    GROUP BY 1
```

pageview_url	total_sessions	total_orders	CVR
/home	2261	72	3.1844
/lander-1	2316	94	4.0587

**Lander-1 has 0.008743 additional orders per sessions compared to home.**

Now, by counting the number of sessions which had happened after the last home page session and multiplying it with the number above we can get the number of orders converted with the newly launched '/lander-1' page.

```
-- Now, finding out the last session_id where home was seen.
SELECT
    MAX(ws.website_session_id)
FROM website_sessions ws
LEFT JOIN
    website_pageviews wp
ON ws.website_session_id = wp.website_session_id
WHERE
    wp.pageview_url = '/home'
    AND ws.created_at < '2012-11-27'
    AND ws.utm_source = 'gsearch'
    AND ws.utm_campaign = 'nonbrand'
-- Most recent home search session 17145.
-- now, finding out how many sessions had happened after 17145 and before assignment date.
SELECT
    COUNT(DISTINCT ws.website_session_id)
FROM website_sessions ws
LEFT JOIN
    website_pageviews wp
ON ws.website_session_id = wp.website_session_id
WHERE
    wp.pageview_url = '/lander-1' AND
    ws.created_at < '2012-11-27'
    AND ws.utm_source = 'gsearch'
    AND ws.utm_campaign = 'nonbrand'
    AND ws.website_session_id > '17145'
```



- After the last home web session, total 22972 sessions had happened till 27/11.
- so, total  $(22972 * 0.008743)$  orders were converted with the lander-1 which is 200 more orders.

**5) For the landing page test, analyzed previously, Let's show a full conversion funnel from each of the two pages to orders. I will use the same period analyzed last time (Jun 19 – Jul 28).**

Conversion funnel: Conversion funnel analysis is about understanding and optimizing each step of your user's experience on their journey toward purchasing your products.

```
-- now, creating primary funnel table
CREATE TEMPORARY TABLE funnel
SELECT
    website_session_id,
    MAX(to_home) AS home_page_As_landing,
    MAX(to_lander) AS lander_page_As_landing,
    MAX(to_products) AS made_it_to_products,
    MAX(to_fuzzy) AS made_it_to_fuzzy,
    MAX(to_cart) AS made_it_to_cart,
    MAX(to_ship) AS made_it_to_ship,
    MAX(to_bill) AS made_it_to_bill,
    MAX(to_thank_you_page) AS made_it_to_thank_you
FROM
    (SELECT
        ws.website_session_id,
        wp.pageview_url,
        CASE WHEN pageview_url = '/home' THEN 1 ELSE 0 END AS to_home,
        CASE WHEN pageview_url = '/lander-1' THEN 1 ELSE 0 END AS to_lander,
        CASE WHEN pageview_url = '/Products' THEN 1 ELSE 0 END AS to_products,
        CASE WHEN pageview_url = '/the-original-mr-fuzzy' THEN 1 ELSE 0 END AS to_fuzzy,
        CASE WHEN pageview_url = '/cart' THEN 1 ELSE 0 END AS to_cart,
        CASE WHEN pageview_url = '/shipping' THEN 1 ELSE 0 END AS to_ship,
        CASE WHEN pageview_url = '/billing' THEN 1 ELSE 0 END AS to_bill,
        CASE WHEN pageview_url = '/thank-you-for-your-order' THEN 1 ELSE 0 END AS to_thank_you_page
    FROM
        website_sessions ws
    LEFT JOIN
        website_pageviews wp
    ON
        ws.website_session_id = wp.website_session_id
    WHERE
        ws.utm_source = 'gsearch' AND
        ws.created_at > '2012-06-19' AND
        ws.created_at < '2012-07-28' AND
        ws.utm_campaign = 'nonbrand' AND
        ws.utm_source = 'gsearch'
    ORDER BY 1) AS flagged_table
GROUP BY 1
```

Now, Let's summarize above table to count further pages for both landing pages.

```
SELECT
  CASE
    WHEN home_page_As_landing = 1 THEN 'Saw_home'
    WHEN lander_page_As_landing = 1 THEN 'Saw_custom_lander'
    ELSE NULL END AS segment,
    COUNT(DISTINCT website_session_id) AS sessions,
    COUNT(DISTINCT CASE WHEN made_it_to_products = 1 THEN website_session_id ELSE null END )/COUNT(
DISTINCT website_session_id) AS to_products,
    COUNT(DISTINCT CASE WHEN made_it_to_fuzzy = 1 THEN website_session_id ELSE null END )/COUNT(
DISTINCT website_session_id) AS to_fuzzy ,
    COUNT(DISTINCT CASE WHEN made_it_to_cart = 1 THEN website_session_id ELSE null END )/COUNT(
DISTINCT website_session_id) AS to_cart,
    COUNT(DISTINCT CASE WHEN made_it_to_ship = 1 THEN website_session_id ELSE null END )/COUNT(
DISTINCT website_session_id) AS to_ship,
    COUNT(DISTINCT CASE WHEN made_it_to_bill = 1 THEN website_session_id ELSE null END )/COUNT(
DISTINCT website_session_id) AS to_bill,
    COUNT(DISTINCT CASE WHEN made_it_to_thank_you = 1 THEN website_session_id ELSE null END )/COUNT(
DISTINCT website_session_id) AS to_thank_you
FROM
  funnel
GROUP BY 1
```

segment	sessions	to_products	to_fuzzy	to_cart	to_ship	to_bill	to_thank_you
Saw_custom_lander	2316	0.4676	0.3333	0.1503	0.0997	0.0851	0.0406
Saw_home	2261	0.4166	0.3025	0.1309	0.0885	0.0743	0.0318



- 6) Let's quantify the impact of our billing test, as well. I will analyze the lift generated from the billing test after releasing new billing page (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.

New, billing test was launched in sept 10, so we will calculate the lift generated in terms of revenue for two months period which is till November 10.

```
SELECT pageview_url,
       COUNT(DISTINCT website_session_id),
       COUNT(DISTINCT order_id),
       SUM(price_usd)/COUNT(DISTINCT website_session_id) AS revenue_per_session
FROM (
  SELECT
    wp.pageview_url,
    wp.website_session_id,
    os.order_id,
    os.price_usd
  FROM
    website_pageviews wp
  LEFT JOIN
    orders os
  ON
    wp.website_session_id = os.website_session_id
  WHERE
    wp.created_at >= '2012-09-10' AND
    wp.created_at < '2012-11-10' AND
    wp.pageview_url IN( '/billing', '/billing-2')) AS billing_order
GROUP BY 1
|
```

pageview_url	COUNT(DISTINCT website_session_id)	COUNT(DISTINCT order_id)	revenue_per_session
/billing	657	300	22.826484
/billing-2	654	410	31.339297

lift generated by billing-2 compared to billing in each month period is 8.51 USD.

now counting total number of sessions with billing page hit to find out impact in revenue of last month (27-10 to 27-11).

```
SELECT
  COUNT(DISTINCT website_session_id) AS sessions
FROM website_pageviews
WHERE pageview_url IN( '/billing', '/billing-2') AND
      created_at < '2012-11-27' AND
      created_at > '2012-10-27'
```

Total sessions when somebody hit the billing page in the last month is 1193.

So, the revenue generated with the lift is  $1193 * 8.51 = 10152$  USD.