E-commerce website traffic performance analysis for Maven Fuzzy Factory

1. The situation

Maven Fuzzy Factory has been live for approximately eight months, and this report is being prepared to present website traffic performance to the board of directors. This analysis uses SQL to explore user behavior and key traffic trends, offering insights into how the website has grown and performed during its first eight months of operation.

2. Website performance analysis

2.1. Overall performance

Maven Fuzzy Factory website performance showed a clear upward trend in both website traffic and order volume over the 8 month period from March to November 2012. Total website sessions grew by 377%. Similarly, orders increased by 522%. While growth was relatively gradual in the first six months, a significant acceleration occurred in October and November, where both orders and sessions saw their largest month-over-month increases. The sharp rise indicates the impact of successful A/B tests for landing pages and billing pages, which is discussed further in the next sections. Figure 1 highlights the growth trend over the 8-month period. Please refer to appendix 1 for the detailed data.

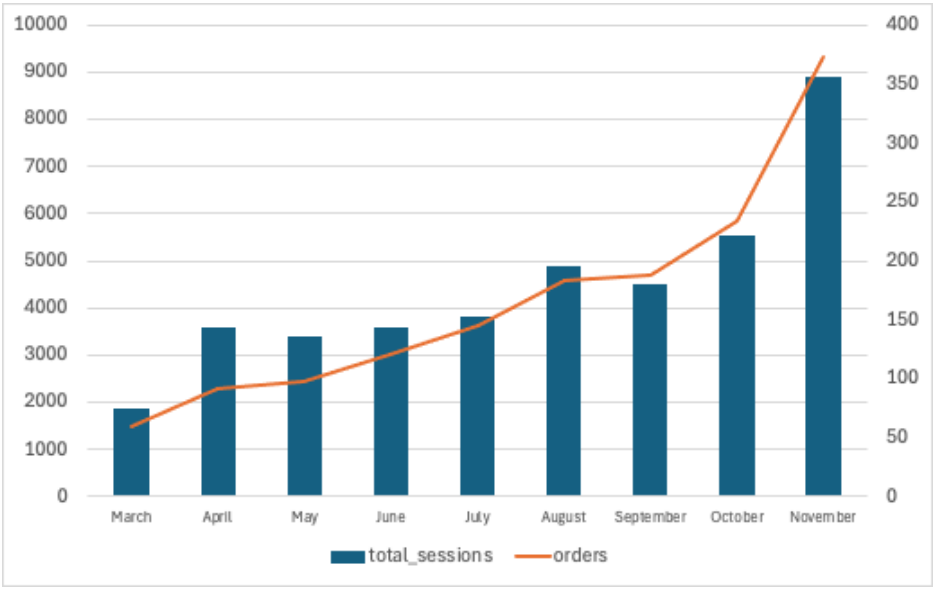

Figure 1. The growth of website performance during the first 8 months of 2012

```sql
-- 1. Monthly trends for gsearch sessions and orders
SELECT
    YEAR(website_sessions.created_at) as year,
    MONTH(website_sessions.created_at) as month,
    COUNT(DISTINCT website_sessions.website_session_id) total_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'
    AND website_sessions.utm_source = 'gsearch'
GROUP BY 1,2;
```

Figure 2. SQL code to extract monthly growth of sessions and orders

2.2 Monthly traffic trend for all channels

The traffic to Maven Fuzzy Factory's website over the past eight months has been driven by four primary channels: Google paid search (gsearch), Bing paid search (bsearch), organic search, and direct type-in sessions. The data shows notable shifts in channel contributions over time. Figure 3 provides a detailed breakdown of monthly traffic by channel, highlighting the relative growth of each source over time.

month	gsearch_paid_sessions	bsearch_paid_sessions	organic_search_sessions	direct_typein_sessions
March	1860	2	8	9
April	3574	11	78	71
May	3410	25	150	151
June	3578	25	190	170
July	3811	44	207	187
August	4877	705	265	250
September	4491	1439	331	285
October	5534	1781	428	440
November	8889	2840	536	485

Figure 3. Monthly traffic trend from March to November 2012

Gsearch consistently remained the dominant source of traffic throughout the period. The steady growth highlights a strong performance of Google Ads in acquiring visitors and customers. Bsearch saw minimal activity in the early months, but experienced a major spike starting in August. Organic Search traffic showed steady and organic growth. Although the absolute numbers remain modest compared to paid channels, the upward trend indicates improved SEO performance or increased brand visibility. Lastly, direct type-in sessions, representing users who accessed

directly through typing URL, also increased gradually. This trend could reflect growing brand recognition and returning visitors, particularly in the last few months.

```sql
-- get a list of all sources and referers of the traffic
SELECT DISTINCT
    utm_source,
    utm_campaign,
    http_referer
FROM website_sessions
WHERE created_at < '2012-11-27';
-- There are several sources of traffic: gsearch paid, bsearch paid, organic search, and direct type-in sessions.
SELECT
    YEAR(website_sessions.created_at) as year,
    MONTH(website_sessions.created_at) as month,
    COUNT(DISTINCT CASE WHEN website_sessions.utm_source='gsearch'
        THEN website_sessions.website_session_id ELSE NULL END) AS gsearch_paid_sessions,
    COUNT(DISTINCT CASE WHEN website_sessions.utm_source='bsearch'
        THEN website_sessions.website_session_id ELSE NULL END) AS bsearch_paid_sessions,
    COUNT(DISTINCT CASE WHEN website_sessions.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 website_sessions.utm_source IS NULL AND http_referer IS NULL
        THEN website_sessions.website_session_id ELSE NULL END) AS direct_typein_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;
-- In particular, organic search and direct type-in sessions have increased over month
```

Figure 4. SQL code to gather all channels' traffic trends

2.3 A/B tests to improve traffic and sales

Landing page test

From June 19 to July 28 2012, Maven Fuzzy Factory initiated an A/B test to evaluate the effectiveness of two landing pages: /home and /lander-1 for gsearch source. The goal was to determine which version led to higher user conversion.

As shown in Figure 5, the two variants attracted a similar number of sessions — 2,261 for /home and 2,316 for /lander-1. However, the outcomes in terms of orders reveal a notable difference: /home resulted in 72 orders (conversion rate: 3.18%), while /lander-1 drove 94 orders, achieving a conversion rate of 4.06%. This result indicates that /lander-1 outperformed /home by a relative conversion lift of approximately 28%, making it the more effective landing page.

landing_page	sessions	orders	conversion_rate
/home	2261	72	0,0318
/lander-1	2316	94	0,0406

Figure 5. A/B Test Results: Performance Comparison of /home vs. /lander-1 landing Pages

Since the implementation of the new landing page, it has contributed to an estimated 202 additional orders over roughly four months, driven by 22,972 extra sessions. This

translates to around 50 extra orders per month, or 0.0088 incremental orders per session.

To further support the results of the A/B test, a full conversion funnel analysis was conducted to track user behavior from the landing page through to order completion.

segment	product_clickthrough_rate	mr_fuzzy_clickthrough_rate	cart_clickthrough_rate	delivery_clickthrough_rate	billing_clickthrough_rate	thankyou_clickthrough_rate
saw_homepage	0,4166	0,7261	0,4327	0,6757	0,84	0,4286
saw_lander	0,4676	0,7128	0,4508	0,6638	0,8528	0,4772

Figure 6. Conversion Funnel Metrics: Users Landing on /home vs. /lander-1

As shown in Figure 6, users who landed on /lander-1 consistently demonstrated slightly higher or comparable clickthrough rates across each step of the funnel, from product page views to final order confirmation. Users who landed on /lander-1 had slightly higher product page clickthrough (46.76% vs. 41.66%), cart click through (45.08% vs. 43.27%), and thank-you page conversion rates (4.77% vs. 4.29%) compared to those who landed on /home. The improved performance across nearly all stages reinforces the decision to adopt /lander-1 as the preferred landing experience.
Please refer to appendix 2 for the full SQL code to conduct the analysis in this part.

Billing page test

To evaluate potential growth opportunities, a new billing page design was tested. During the final month of the analysis period (October 27 to November 27, 2012), a clear improvement in sales conversion was observed. The old billing page generated an average of $22.83 in revenue per billing page view, while the new version generated $31.34, reflecting an increase of $8.51 per view.

Out of 1,311 total sessions during this period, 1,193 sessions reached the billing page, resulting in an estimated incremental value of $10,160 attributable to the new billing page design.

Please refer to figure 7 for the result and figure 8 for SQL code.

billing_version_seen	sessions	revenue_per_billing_page_seen
/billing	657	22,826484
/billing-2	654	31,339297

Figure 7. Revenue Performance by Billing Page Version (Oct 27 - Nov 27, 2012)

```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'
    AND website_pageviews.created_at < '2012-11-10'
    AND website_pageviews.pageview_url IN ('/billing','/billing-2')) AS billing_pageviews_and_order_data
GROUP BY 1;

-- calculate the total amount of billing sessions
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'
```

Figure 8: SQL code for revenue comparison between billing page versions

APPENDIX

Appendix 1. General growth of sessions, orders, and conversion rate

year	month	total_sessions	orders	conversion_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	0,038
2012	8	4877	184	0,0377
2012	9	4491	188	0,0419
2012	10	5534	234	0,0423
2012	11	8889	373	0,042

Appendix 2. SQL code for Landing page A/B test analysis

1. **Estimate revenue impact of the landing page test**

```
-- Step 1: Identify the first pageview ID for the test
SELECT MIN(website_pageview_id) AS first_test_pv
FROM website_pageviews
WHERE pageview_url = '/lander-1';

-- Step 2: Confirm the date of the first pageview in the test
SELECT created_at
FROM website_pageviews
WHERE website_pageviews.website_pageview_id >= 23504;

-- Step 3: Create temporary table of first pageview per session
CREATE TEMPORARY TABLE first_page_viewed AS
SELECT
        website_pageviews.website_session_id,
   MIN(website_pageviews.website_pageview_id) AS min_pageview_id
FROM website_pageviews
JOIN website_sessions ON website_sessions.website_session_id =
website_pageviews.website_session_id
WHERE website_sessions.created_at < '2012-07-28'
  AND website_pageviews.website_pageview_id >= 23504
  AND utm_source = 'gsearch'
  AND utm_campaign = 'nonbrand'
GROUP BY website_pageviews.website_session_id;
```

```
-- Step 4: Attach landing page URL to each session
CREATE TEMPORARY TABLE nonbrand_test_sessions_with_landingpages AS
SELECT
        first_page_viewed.website_session_id,
   first_page_viewed.min_pageview_id,
   website_pageviews.pageview_url AS landing_page
FROM first_page_viewed
LEFT JOIN website_pageviews
  ON first_page_viewed.min_pageview_id =
website_pageviews.website_pageview_id
WHERE website_pageviews.pageview_url IN ('/home', '/lander-1');

-- Step 5: Add order data
CREATE TEMPORARY TABLE nonbrand_test_sessions_w_orders AS
SELECT
        nonbrand_test_sessions_with_landingpages.website_session_id,
   nonbrand_test_sessions_with_landingpages.landing_page,
   orders.order_id
FROM nonbrand_test_sessions_with_landingpages
LEFT JOIN orders
  ON orders.website_session_id =
nonbrand_test_sessions_with_landingpages.website_session_id;

-- Step 6: Calculate conversion rate by landing page
SELECT
        landing_page,
   COUNT(DISTINCT website_session_id) AS sessions,
   COUNT(DISTINCT order_id) AS orders,
   COUNT(DISTINCT order_id) * 1.0 / COUNT(DISTINCT website_session_id) AS
conversion_rate
FROM nonbrand_test_sessions_w_orders
GROUP BY landing_page;
```

2. **Estimate incremental orders from the test**

```
-- Step 7: Find the last session using the old landing page
SELECT MAX(website_sessions.website_session_id) AS
most_recent_gsearch_nonbrand_home_pv
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 < '2012-11-07';
```

```
-- Step 8: Count the sessions since the switch to new landing page
SELECT COUNT(website_session_id) AS sessions_since_test
FROM website_sessions
WHERE created_at < '2012-11-27'
  AND website_session_id > 17145  -- based on above query
  AND utm_source = 'gsearch'
  AND utm_campaign = 'nonbrand';
```

3. Full Conversion Funnel Analysis

```
-- Step 9: Flag all key steps in the funnel for each session
CREATE TEMPORARY TABLE sessions_level_flagged AS
SELECT
        website_session_id,
    MAX(homepage) AS homepage_click,
    MAX(lander_page) AS lander_click,
    MAX(products_page) AS product_click,
    MAX(fuzzy_page) AS fuzzy_click,
    MAX(cart_page) AS cart_click,
    MAX(delivery_page) AS delivery_click,
    MAX(billing_page) AS billing_click,
    MAX(thankyou_page) AS thankyou_click
FROM (
  SELECT
        website_sessions.website_session_id,
    website_pageviews.website_pageview_id,
    CASE WHEN pageview_url = '/home' THEN 1 ELSE 0 END AS homepage,
    CASE WHEN pageview_url = '/lander-1' THEN 1 ELSE 0 END AS lander_page,
    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
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 delivery_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_pageviews.website_session_id =
website_sessions.website_session_id
  WHERE website_sessions.created_at BETWEEN '2012-06-19' AND '2012-07-28'
    AND utm_source = 'gsearch'
    AND utm_campaign = 'nonbrand'
) AS flag_list
GROUP BY website_session_id;
-- Step 10: Count step completions by landing page group
CREATE TEMPORARY TABLE session_click_to_flagged AS
```

```
SELECT
  CASE
    WHEN homepage_click = 1 THEN 'saw_homepage'
    WHEN lander_click = 1 THEN 'saw_lander'
    ELSE 'error'
  END AS segment,
  COUNT(DISTINCT website_session_id) AS sessions,
  COUNT(DISTINCT CASE WHEN product_click = 1 THEN website_session_id
END) AS to_product,
  COUNT(DISTINCT CASE WHEN fuzzy_click = 1 THEN website_session_id END)
AS to_mr_fuzzy,
  COUNT(DISTINCT CASE WHEN cart_click = 1 THEN website_session_id END)
AS to_cart,
  COUNT(DISTINCT CASE WHEN delivery_click = 1 THEN website_session_id
END) AS to_delivery,
  COUNT(DISTINCT CASE WHEN billing_click = 1 THEN website_session_id END)
AS to_billing,
  COUNT(DISTINCT CASE WHEN thankyou_click = 1 THEN website_session_id
END) AS to_thankyou
FROM sessions_level_flagged
GROUP BY 1;

-- Step 11: Calculate clickthrough rates across funnel stages
SELECT
        segment,
    to_product * 1.0 / sessions AS product_clickthrough_rate,
    to_mr_fuzzy * 1.0 / to_product AS mr_fuzzy_clickthrough_rate,
    to_cart * 1.0 / to_mr_fuzzy AS cart_clickthrough_rate,
    to_delivery * 1.0 / to_cart AS delivery_clickthrough_rate,
    to_billing * 1.0 / to_delivery AS billing_clickthrough_rate,
    to_thankyou * 1.0 / to_billing AS thankyou_clickthrough_rate
FROM session_click_to_flagged;
```