Analysis of CoolTShirt's Marketing Attribution

Analyzing Touch Attribution with SQL

by Marie Gomez

Data provided by codecademy

Table of Contents

1. Overview

Project Overview

The purpose of this project is to analyze marketing attributions at CoolTShirts (CTS). Throughout the process, I utilize SQL, UTM parameters, & first- and last-touch attributions to gather insight for the company, and conclude with recommendations based on analyses conducted.

Company Background:

CoolTShirts, is an innovative e-commerce shirt shop running a number of marketing campaigns to increase website visits and purchases.

The marketing team is interested in analyzing their customers' journey (from initial visit to purchase) to optimize their marketing campaign budget.

Dataset provided in this project:

1 SQL table called 'page_visits'

- Contains 5,692 record
- 5 columns:
  - page_name – the title of the sections of the page that was visited
  - timestamp – the time at which the visitor came to the page
  - user_id – a unique id for each visitor
  - utm_campaign – identity of the specific ad or email blast
  - utm_source – identity of which touchpoint sent the traffic (e.g. google, email, fb, etc)

2. Inspect the Data

Inspect the Data

CTS has four main pages on their website.

1. landing_page
2. shopping_cart
3. checkout
4. purchase

The company is using eight distinct campaigns and six distinct sources.

To the right, the last query indicates which source is used for each campaign.

Example: marketing campaign ‘getting-to-know-cool-shirts’ was clicked on by a user who was previously on the nytimes website.

This example indicates that the *first touch* – the first time a user was exposed to CoolTShirts.com - was attributed to the nytimes as the source.

```sql
SELECT *
FROM page_visits
LIMIT 20;
SELECT COUNT(*)
FROM page_visits;

SELECT COUNT(DISTINCT page_name) AS num_pages
FROM page_visits;

SELECT COUNT(DISTINCT utm_campaign) AS num_campaigns
FROM page_visits;

SELECT COUNT(DISTINCT utm_source) AS num_sources
FROM page_visits;

SELECT DISTINCT utm_campaign, utm_source
FROM page_visits;
```

num_pages
4
num_campaigns
8
num_sources
6

utm_campaign	utm_source
getting-to-know-cool-tshirts	nytimes
weekly-newsletter	email
ten-crazy-cool-tshirts-facts	buzzfeed
retargetting-campaign	email
retargetting-ad	facebook
interview-with-cool-tshirts-founder	medium
paid-search	google
cool-tshirts-search	google

page_name	timestamp	user_id	utm_campaign	utm_source
1 - landing_page	2018-01-24 03:12:16	10006	getting-to-know-cool-tshirts	nytimes
2 - shopping_cart	2018-01-24 04:04:16	10006	getting-to-know-cool-tshirts	nytimes
3 - checkout	2018-01-25 23:10:16	10006	weekly-newsletter	email
1 - landing_page	2018-01-25 20:32:02	10030	ten-crazy-cool-tshirts-facts	buzzfeed
2 - shopping_cart	2018-01-25 23:05:02	10030	ten-crazy-cool-tshirts-facts	buzzfeed
3 - checkout	2018-01-28 13:26:02	10030	retargetting-campaign	email
4 - purchase	2018-01-28 13:38:02	10030	retargetting-campaign	email

3. What is the User Journey?

3.1 What is the User Journey?

Aggregate the number of touch attributes each campaign is responsible for:

The *first touch* queries the number of users who were the first time exposed to CTS.com.

- Utm campaigns with the most first touches:
    1. 'interview-with-cool…'
    2. 'getting-to-know…'
    3. 'ten-crazy-cool…'

The *last touch* queries the number of users who – after the exposure to CTS.com - made a purchase.

- Utm campaigns with the most last touches:
    1. 'weekly-newsletter'
    2. 'retargeting-ad'
    3. 'retargeting-campaign'
    4. 'getting-to-know-cool…'

utm_campaign	num_first_touches
interview-with-cool-tshirts-founder	622
getting-to-know-cool-tshirts	612
ten-crazy-cool-tshirts-facts	576
cool-tshirts-search	169

```sql
/*How many first-touches is each campaign
responsible for?*/
WITH first_touch AS (
    SELECT user_id,
        MIN(timestamp) as first_touch_at
    FROM page_visits
    GROUP BY user_id)
SELECT
    pv.utm_campaign,
    COUNT(utm_campaign) as num_first_touches
FROM first_touch AS ft
JOIN page_visits AS pv
    ON ft.user_id = pv.user_id
    AND ft.first_touch_at = pv.timestamp
GROUP BY utm_campaign
ORDER BY count(utm_campaign) DESC;
```

utm_campaign	num_last_touches
weekly-newsletter	447
retargetting-ad	443
retargetting-campaign	245
getting-to-know-cool-tshirts	232
ten-crazy-cool-tshirts-facts	190
interview-with-cool-tshirts-founder	184
paid-search	178
cool-tshirts-search	60

```sql
/*How many last-touches is each campaign responsible
for?*/
WITH last_touch AS (
  SELECT user_id,
    MAX(timestamp) as last_touch_at
  FROM page_visits
  GROUP BY user_id)
SELECT
  pv.utm_campaign,
  count(utm_campaign) as num_last_touches
FROM last_touch as lt
JOIN page_visits as pv
  ON lt.user_id = pv.user_id AND
  lt.last_touch_at = pv.timestamp
GROUP BY utm_campaign
ORDER BY count(utm_campaign) DESC;
```

3.2 What is the User Journey?

How many visitors make a purchase?

What does the overall purchase funnel look like for each unique visitor that makes to the landing page and eventually makes a purchase on CTS.com?

page_name	num_visitor	overall_funnel_%
1 - landing_page	1979	100.0
2 - shopping_cart	1881	95.0
3 - checkout	1431	72.0
4 - purchase	361	18.0

Key Findings:

It begins with 1,979 unique visitors making it to the landing page. Of those 1,979 visitors it funnels down to only 361 visitors who make it to the purchase page to finalize a purchase.

18% of the original unique visitors decided to make a purchase.

```
/*How many visitors make a purchase?*/
SELECT
  page_name,
  COUNT(DISTINCT user_id) AS 'num_visitor'
FROM page_visits
GROUP BY page_name;
/*How many visitors make a purchase in the form of an
overall Funnel percentage?*/
WITH page_visit_total AS (
  SELECT
  page_name,
  COUNT(DISTINCT user_id) AS 'num_visitor'
FROM page_visits
GROUP BY page_name
)
SELECT page_name, num_visitor,
  ROUND(100*(CAST(num_visitor AS REAL)/1979)) AS
'overall_funnel_%'
FROM page_visit_total
GROUP BY page_name;
```

3.3 What is the User Journey?

How many last touches on the purchase page is each campaign responsible for?

utm_campaign	page_name	num_campaigns
weekly-newsletter	4 - purchase	115
retargetting-ad	4 - purchase	113
retargetting-campaign	4 - purchase	54
paid-search	4 - purchase	52
ten-crazy-cool-tshirts-facts	4 - purchase	9
getting-to-know-cool-tshirts	4 - purchase	9
interview-with-cool-tshirts-founder	4 - purchase	7
cool-tshirts-search	4 - purchase	2

```sql
WITH last_touch AS (
  SELECT user_id,
    MAX(timestamp) as last_touch_at
  FROM page_visits
  WHERE page_name == '4 - purchase'
  GROUP BY user_id)
SELECT
  pv.utm_campaign, page_name,
  count(utm_campaign) as num_campaigns
FROM last_touch as lt
JOIN page_visits as pv
  ON lt.user_id = pv.user_id AND
  lt.last_touch_at = pv.timestamp
GROUP BY utm_campaign
ORDER BY count(utm_campaign) DESC;
```

Key Findings:

Query results show the top three campaigns that lead visitors back to CTS.com (from elsewhere) to finalize their purchase (from where they left off on the purchase page) were:

1. ‘weekly-newsletter’
2. ‘retargeting-ad’
3. ‘retargeting-campaign’

4. Recommendations

Recommendations

To optimize the campaign budget CoolTShirts can re-invest in 5 campaigns.

It is recommended that CTS.com select these five campaigns:

1. interview-with-cool-tshirts-founder. *(first-touch)*
2. getting-to-know-cool-tshirts *(first-touch)*
3. ten-crazy-cool-tshirts-facts *(first-touch)*
4. weekly-newsletter. *(last-touch)*
5. retargeting-ad. *(last-touch)*

Based on the queries aggregating *first-touches*, campaigns 1,2, & 3 lead to the most exposure.

For *last-touches*, campaigns 4 & 5 showed the highest results in bringing visitors back to CTS.com to finalize their purchases.