codecademy

Capstone Project-Calculating Churn Rates

Learn SQL from Scratch

Table of Contents

1. Get familiar with Codeflix

1.1 Codeflix - Segments and Months

Codeflix has decided to analyze 2 methods of acquiring and maintaining users

- There are 2 segments available to analyze
- Each has 1000 distinct users over a 4 month period
- Churn analysis doesn't apply to the first month as we would have 0 active users to begin with

```sql
SELECT min(subscription_end) AS earliest_end,
min(subscription_start) AS earliest_start,
max(subscription_end) AS latest_end,
max(subscription_start) AS latest_start
FROM subscriptions;
```

earliest_end	earliest_start	latest_end	latest_start
2017-01-01	2016-12-01	2017-03-31	2017-03-3

```sql
select count(DISTINCT id) as users,
segment
from subscriptions
group by segment;
```

users	segment
1000	87
1000	30

2. What is the overall churn trend

2.1 Defining Months

In order to calculate churn, we will define active users at the beginning of a month and users who canceled in that month.

We will create a temporary table called months to define the range of dates that fall into each month

first_day	last_day
2017-01-01	2017-01-31
2017-02-01	2017-02-28
2017-03-01	2017-03-31

```
WITH months AS
(
SELECT
  '2017-01-01' AS first_day,
  '2017-01-31' AS last_day
UNION
  SELECT
  '2017-02-01' AS first_day,
  '2017-02-28' AS last_day
UNION
  SELECT
  '2017-03-01' AS first_day,
  '2017-03-31' AS last_day
)
SELECT *
FROM months
LIMIT 3;
```

2.2 Checking Subscription Status

Next, we will use a cross join to combine our subscription data with our months table.

- This makes a new, very large table as every id is paired with each month.
- This will allow us to compare a user's start and end date with the start and end date for each month

```
14  ),
15  cross_join AS (
16    SELECT *
17  FROM subscriptions
18  CROSS JOIN months
19  )
20  SELECT *
21  FROM cross_join
22  LIMIT 4;
```

id	subscription_start	subscription_end	segment	first_day	last_day
1	2016-12-01	2017-02-01	87	2017-01-01	2017-01-31
1	2016-12-01	2017-02-01	87	2017-02-01	2017-02-28
1	2016-12-01	2017-02-01	87	2017-03-01	2017-03-31
2	2016-12-01	2017-01-24	87	2017-01-01	201-01-03

2.3 Calculating The Churn Trend

In order to calculate the churn trend, we will create a final table “status” that will use a case structure to mark users as active or canceled.

- After applying the case structure, we group the users by segment and month so that is easier to view all the churn rates at once

segment	month	churn_rate	active_users
30	2017-01-01	7.5601	291
30	2017-02-01	7.3359	518
30	2017-03-01	11.7318	716
87	2017-01-01	25.1799	278
87	2017-02-01	32.0346	462
87	2017-03-01	48.5876	531

```
status AS (
SELECT id,
  segment,
  first_day as month,
  CASE
      WHEN (subscription_start < first_day)
      AND
      (subscription_end > first_day OR subscription_end IS NULL)
      THEN 1
      ELSE 0
      END AS is_active,
  CASE
      WHEN (subscription_end BETWEEN first_day AND last_day)
      THEN 1
      ELSE 0
      END AS is_canceled
  FROM
  cross_join
)
 SELECT segment,
 month,
 ROUND(100.0*SUM(is_canceled)/SUM(is_active),4) AS churn_rate,
 SUM(is_active) AS active_users
 FROM status
 GROUP BY segment, month
 ORDER BY segment, month;
```

3. Churn Analysis

3.1 Analysis and Recommendations

- Our analysis indicates that segment 30 has a much **smaller churn rate** as well as a **higher active user count**.
- Churn rates increased for users acquired by both segments as time went on, but the change was negligible for segment 30 users.
- The **user base for both methods continues to grow**, but segment 87 users are almost leaving as quickly as they are joining.
- We would recommend **increasing investment** in segment 30 advertising.
- However, around 40% of your users were still connected with through segment 87. We would recommend c**ontinuing to invest** in segment 80 advertising **to maintain your user base**.