codecademy

Codeflix Churn Trends

By Nick Massa

Learn SQL from Scratch

Table of Contents

1. Codeflix Operations

1.1 Company Operations

- Codeflix has been operating for four months from December 1st, 2016 to March 31st, 2017.
- We are able to calculate churn rate only for three months, January 2017 to March 2017 as the earliest subscription end date is January 1st, 2017.

Most Recent Start	Oldest Start	Most Recent End	Oldest End
2017-03-30	2016-12-01	2017-03-31	2017-01-01

```
select
max(subscription_start) as 'Most Recent Start',
min(subscription_start) as 'Oldest Start',
max(subscription_end) as 'Most Recent End',
min(subscription_end) as 'Oldest End'
from subscriptions;
```

1.2 Company Operations

- Codeflix has two segments of users, 30 and 87.
- Each segment has 1000 users

```
select segment, count(segment) as 'rows per segment'
from subscriptions group by segment;
```

segment	rows per segment	
30	1000	
87	1000	

2. Codeflix Churn Rates by Month

2.1 Churn Rates by Month

- Codeflix as a whole has seen an increase in the Churn rate month over month
- To derive this, we created a temporary table of months, cross joined that to the subscriptions table, created columns to count if a subscriber is active or canceled in each month, totaled all of these columns by month and segment, then divided the active totals by the canceled totals.

month	Seg87_churn	Seg30_churn
2017-01-01	0.251798561151079	0.0756013745704467
2017-02-01	0.32034632034632	0.0733590733590734
2017-03-01	0.485875706214689	0.11731843575419

```
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),

cross_join as
 (select * from subscriptions cross join months),

 status as (select id, first_day as month,
case when (segment = 87) and (subscription_start <
first_day) and (subscription_end > first_day or
subscription_end is null) then 1 else 0 end as is_active_87,
case when (segment = 30) and (subscription_start <
first_day) and (subscription_end > first_day or
subscription_end is null) then 1 else 0 end as is_active_30,
case when (segment = 87) and (subscription_end between
first_day and last_day) then 1 else 0 end as is_canceled_87,
case when (segment = 30) and (subscription_end between
first_day and last_day) then 1 else 0 end as is_canceled_30
from cross_join),

status_aggregate as (select month,
  sum(is_active_87) as 'sum_active_87',
  sum(is_active_30) as 'sum_active_30',
  sum(is_canceled_87) as 'sum_canceled_87',
  sum(is_canceled_30) as 'sum_canceled_30'
  from status
  group by month
  order by month)

select month, 1.0 * sum_canceled_87 / sum_active_87 as
'Seg87_churn', 1.0 * sum_canceled_30 / sum_active_30 as
'Seg30_churn'
from status_aggregate;
```

3. Codeflix Churn Rates by Segment and Future Recommendations

3.1 Churn Rates by Segment

- The churn trend on Segment 30 has increased relatively slightly over the three month time period from 7% in January and February, to 11% in march.
- The churn trend on segment 87 has increased much more dramatically over the three month period from 25% to 32% to 48%.

month	Seg87_churn	Seg30_churn
2017-01-01	0.251798561151079	0.0756013745704467
2017-02-01	0.32034632034632	0.0733590733590734
2017-03-01	0.485875706214689	0.11731843575419

3.2 Recommendations

Going forward Codeflix should:

- Focus on expanding Segment 30 as they have a much lower churn trend on those users and thus lower cost to acquire and retain users.
- Research the large jump in churn on Segment 87 in order to devise a way to curb Codeflix cancelations.
- Create more user segments so they can more easily track and attribute Churn to specific factors.