codecademy

Battling Churn at Codeflix

Capstone: Churn Rates

Table of Contents

1. Get familiar with Codeflix

1.1 Streaming since December 2016

Codeflix is a video streaming startup, that has been operating since December 2016. This can be seen from the lowest subscription_start date in the ‘subscriptions table’, which is December 1st, 2016. There are no ended subscriptions December 2016. The last subscription started March 30th 2017. Therefore, we’ll be able to calculate the churn rate for the months of January, February and March.

first_subscription	last_subscription
2016-12-01	2017-03-30

```
SELECT
      MIN(subscription_start) AS first_subscription,
      MAX(subscription_start) AS last_subscription
FROM subscriptions;
```

1.2 The first hundred rows: Segmenting

The data in the ‘subscription’ table not only gives us information about when a user started and ended his subscription, but also about the segment he has been assigned to.
The first hundred rows of ‘subscription’ shows us, that the company has been segmenting its users into two segments: 87 and 30. Querying all distinct segment rows firms this conclusion.

id	subscription_start	subscription_end	segment
11	2016-12-01	2017-01-17	87
12	2016-12-01	2017-02-07	87
13	2016-12-01		30
14	2016-12-01	2017-03-07	30
15	2016-12-01	2017-02-22	30

2. What is the overall churn trend since the company started?

2.1 Overall Churn Rate

The overall Codeflix churn rate is 22.17 %.

But: That doesn't tell us much about the churn trend. We'll have to look into the monthly churn rates, to see how it change during this three months.

```
WITH months AS
 (...),
cross_join AS
 (...),
status AS
 (...),
status_aggregate AS
       (SELECT
         SUM(is_active) AS sum_active,
         SUM(is_canceled) AS sum_canceled
       FROM status)
SELECT
 ROUND(1.0*sum_canceled/sum_active, 4) AS churn_rate
FROM status_aggregate;
```

2.2 Monthly Churn Rate

Now that we have the monthly churn rate, there are some bad news: The churn rate is increasing. In March it even exceeded the overall churn rate of 22.17 %.

I recommend digging deeper into the data. One hypothesis is that the higher churn_rate could be related to a quarterly payment option. To evaluate this hypothesis we need payment related information, that isn't given in the 'subscriptions' table.

month	churn_rate
2017-01-01	0.1617
2017-02-01	0.1898
2017-03-01	0.2743

```
WITH months AS
 (...),
cross_join AS
 (...),
status AS
 (...),
status_aggregate AS
       (SELECT
         month,
         SUM(is_active) AS sum_active,
         SUM(is_canceled) AS sum_canceled
       FROM status
       GROUP BY month)
SELECT
 month,
 ROUND(1.0*sum_canceled/sum_active, 4) AS churn_rate
FROM status_aggregate;
```

3. Compare the churn rates between segments

3.1 Battle of the Segments

Fortunately, Codeflix segmented its users. Calculating the churn rate of the different segments, we see that segment 87 has 37.45 % whereas segment 30 has only 9.44 %.

That is a huge difference that shows us, that the overall churn rate doesn't tell as much about the behaviour of Codeflix' committed users - located in segment 30.

Codefix should focus on segment 30 and expand the number of users in this segment.

churn_87	churn_30
0.3745	0.0944

```
...
status_aggregate AS
       (SELECT
         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)
SELECT
 ROUND(1.0*sum_canceled_87/sum_active_87, 4) AS
churn_87,
 ROUND(1.0*sum_canceled_30/sum_active_30, 4) AS
churn_30
FROM STATUS_aggregate;
```

3.2 More segments

The results show the importance of segmenting. For the future, I recommend Codeflix to establish more segments and to keep comparing them over time.

With a few changes in the temporary tables ‘status’ and ‘status_aggregate’, we can query the different churn rates of any given amount of segments.

On the right, you can see how the results of the altered query will look like.

segment	churn_rate
30	0.0944
87	0.3745

3.3 Monthly Churn Rate per Segment

If we look into the churn trend of both segments, we see that they match the overall increasing churn trend. This underlines the necessity to dig a bit deeper into the causes of this increasing trend.

month	segment	churn_rate
2017-01-01	30	0.0756
2017-02-01	30	0.0734
2017-03-01	30	0.1173
2017-01-01	87	0.2518
2017-02-01	87	0.3203
2017-03-01	87	0.4859

Battling Churn at Codefix: To-Do’s

- ❏ Expand segment 30: They are committed users!
- ❏ Install more segments to compare and experiment with.
- ❏ Gather more data to find out, why the overall churn is increasing. A deeper look into payment options could help.