codecademy

Capstone Project: Codeflix Churn

Learn SQL from Scratch

By Jacob Nicholson
August 27, 2018

Project Overview

During this presentation, we will be examining data from Codeflix, the latest player in the video streaming scene. Over the course of this examination we will be exploring the following questions:

1. Getting familiar with Codeflix
    1. How many months has Codeflix been operating?
    2. For which months can we analyze a churn rate?
    3. Which segments of users exist?
2. What is the overall churn trend since the company started?
3. Compare the churn rates between segments.
    1. Which segments of users should the company focus on expanding?

codecademy

1. Getting familiar with Codeflix

1. Getting to know Codeflix

A quick query of Codeflix's *subscriptions* data base gives us a sense of their parameters. Currently they are tracking their individual users, those users' start & end dates for their subscription, and which segment they were a part of.

```
SELECT *
FROM subscriptions
LIMIT 100;
```

Id	subscription_start	subscription_end	Segment
1	2016-12-01	2017-02-01	87
2	2016-12-01	2017-01-24	87
3	2016-12-01	2017-03-07	87

1. Getting to know Codeflix, pt. II

Querying the earliest and latest subscription dates, we see that Codeflix has data across 4 months, from December 2016 through March 2017. However, since subscriptions have a minimum of 1 month duration, there will not be any cancellations during December, leaving us with 3 months of valid churn data to analyze.

earliest_sub	latest_sub
2016-12-01	2017-03-30

```
SELECT MIN(subscription_start) AS ‘earliest_sub’,
    MAX(subscription_start) AS ‘latest_sub
FROM subscriptions;
```

1. Getting to know Codeflix, pt. III

Codeflix is interested in examining their churn rate across different segments. In this case, there are only two segments in the database.

Segment
87
30

```
SELECT DISTINCT segment
FROM subscriptions;
```

codecademy

2. What is the overall churn trend?

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

This inquiry is asking about the *overall* churn rate, so the first query lays out the churn rate across the first three months of 2017 without delineating between segments. In addition, although the original churn rates given express in decimal format (1.0 * canceled/active) out to as many as 13 decimal places, this information is presumably being presented to management staff or possibly non-technical team members. As such, in this presentation the churn rate is presented as a percentage (100.0 * canceled/active) rounded to 2 decimal places, so that the result is presented as a more easily understood percent of the total user base. A margin of error of 0.01% due to rounding equates to < 1 user based on their total database of 2000 users. Due to the length of the query, the code is presented on the next slide and the results of such are below.

month	churn_rate_percent
2017-01-01	16.17
2017-02-01	18.98
2017-03-01	27.43

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

```
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 cross_join.id as id,
    cross_join.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),
status_aggregate AS (
  SELECT month,
    sum(is_active) as 'active',
    sum(is_canceled) as 'canceled'
  FROM status
  GROUP BY month)
SELECT
  month,
  ROUND(100.0 * canceled/active,2) AS
'churn_rate_percent'
FROM status_aggregate;
```

codecademy

3. Comparing churn rates between segments.

3. Comparing churn rates between segments

The churn rate can be further separated out between the two segments, showing the churn rate for each segment independently. Again, these rates are presented as percentages rounded out to 2 decimal places for easier reading in a less technical setting. The following shows these churn rates, with the code on the following slide.

month	87_churn_rate	30_churn_rate
2017-01-01	25.18	7.56
2017-02-01	32.03	7.34
2017-03-01	48.59	11.73

3. Comparing churn rates between segments (code)

```
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 cross_join.id as id,
    cross_join.first_day as 'month',
    CASE
      WHEN
        (subscription_start < first_day)
      AND
        (subscription_end > first_day
        OR
        subscription_end IS NULL)
      AND
        (segment = '87')
      THEN 1
      ELSE 0
      END AS is_active_87,
```

```
CASE
      WHEN
        (subscription_start < first_day)
      AND
        (subscription_end > first_day
        OR
        subscription_end IS NULL)
      AND
        (segment = '30')
      THEN 1
      ELSE 0
      END AS is_active_30,
    CASE
      WHEN
        (subscription_end BETWEEN first_day AND
last_day)
      AND
        (segment = '87')
      THEN 1
      ELSE 0
      END AS is_canceled_87,
    CASE
      WHEN
        (subscription_end BETWEEN first_day AND
last_day)
      AND
        (segment = '30')
      THEN 1
      ELSE 0
      END AS is_canceled_30
    FROM cross_join),
```

3. Comparing churn rates between segments (code, cont'd)

```
status_aggregate AS (
  SELECT month,
    sum(is_active_87) as 'active_87',
    sum(is_active_30) as 'active_30',
    sum(is_canceled_87) as 'canceled_87',
    sum(is_canceled_30) as 'canceled_30'
  FROM status
  GROUP BY month)
SELECT
  month,
  ROUND(100.0 * canceled_87/active_87,2) AS
'87_churn_rate',
  ROUND(100.0 * canceled_30/active_30,2) AS
'30_churn_rate'
FROM status_aggregate;
```

3. Which segment of users should the company focus on expanding?

At first glance, the data is fairly clear. The users of segment 87 are dropping out at a much greater rate than those in segment 30, around 3-4 times as much. However a percentage can be misleading. If segment 87 only represents a small portion of the overall user base, the higher percentage may not be so alarming.
A quick query confirms that, in fact, both segments are equal.

total_subs
2000
Segment_87_subs
1000
Segment_30_subs
1000

```
SELECT COUNT(*) AS 'total_subs'
FROM subscriptions;

SELECT COUNT(*) AS 'segment_87_subs'
FROM subscriptions
WHERE segment = 87;

SELECT COUNT(*) AS 'segment_30_subs'
FROM subscriptions
WHERE segment = 30;
```

3. Which segment of users should the company focus on expanding?

From here there is an argument to be made either way, depending on what strategies the company is considering to take. Since segment 30 tends to retain much more subscribers than segment 87, it might be that whatever market or audience that segment is targeting is more receptive to the subscription services. That community might then be a more valuable target for future marketing, advertisement or recruiting campaigns.

However, it can also be argued that since segment 87 loses so many subscribers, that focusing on retention efforts for that segment of customers might also help stabilize the existing user base. If a campaign is introduced that incentivizes users to return, such as promotional subscription rates or special offers, then segment 87 offers the most potential re-subscribers, making them potentially a more effective target audience to recapture.

If all such strategies are currently being deployed that the company is interested in pursuing, then segment 30 does clearly represent the best “stay the course” option to continue in the future, as more users in that segment tend to stay as subscribers and so offers the best long-term user base according to current churn trends.