



Codeflix

Churn Rates - Analyze Data with SQL

Firat Olçum

03-07-2022

Table of Contents

1. Introduction
2. Methods
3. Results & Discussion

1. Introduction

What is Churn Rate?

- Churn rate is the percent of subscribers that have canceled within a certain period, usually a month.
- To calculate the churn rate, we only will be considering users who are subscribed at the beginning of the month.
- The churn rate is the number of these users who cancel during the month divided by the total number.

Codeflix, a streaming video startup, is interested in measuring their user churn rate.

- Four months into launching Codeflix, management asks me to look into subscription churn rates. It's early on in the business and people are excited to know how the company is doing.
- The marketing department is particularly interested in how the churn compares between two segments of users. They provide me with a dataset containing subscription data for users who were acquired through two distinct channels.

Key Questions

1. How many months has the company been operating? Which months do you have enough information to calculate a churn rate?
2. What segments of users exist?
3. What is the overall churn trend since the company started?

Churn rate = cancelled users / active users
4. Compare the churn rates between user segments. Which segment of users should the company focus on expanding?

2. Methods

Basic Information

The dataset provided to me contains one SQL table, `subscriptions`. Within the table, there are 4 columns:

- `id` - the subscription id.
- `subscription_start` - the start date of the subscription.
- `subscription_end` - the end date of the subscription.
- `segment` - this identifies which segment the subscription owner belongs to.

Codeflix requires a minimum subscription length of 31 days, so a user can never start and end their subscription in the same month.

2.1 Get Familiar with the Codeflix data

When I look at the Codeflix subscriptions table, I see that there are two different segments.

Two (2) segments:

- Segment 87
- Segment 30

segment
87
30

```
SELECT
  DISTINCT segment
FROM
  Codeflix_Subscriptions;
```

2.2 Get Familiar with the Codeflix data

I will determine the range of months of data provided. I will look which months will I be able to calculate churn.

Data has a start date range of 2016-12-01 to 2016-03-30. I will be able to calculate churn only for January, February, and March. December doesn't have any active users since a user is defined as active if they signed up before the first of the month. There are also no cancelled users in December since Codeflix requires a minimum 31 day subscription.

start_date	end_date
2016-12-01	2017-03-30

```
SELECT
  MIN(subscription_start) AS start_date,
  MAX(subscription_start) AS end_date
FROM
  Codeflix_Subscriptions;
```

2.3 Calculate Churn for Each Segment: Create a Months Table

Our churn calculation uses the first day as a cutoff for subscribers and the last day as a cutoff for cancellations. I will need the first and last day of each month. I will create table of months in order to make a comparison in the next few steps.

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

```
CREATE TABLE 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;
```

2.4 Calculate Churn for Each Segment: Cross Join subscriptions to months

I will create a table with all possible combinations to set up a comparison that determines what month start dates and end dates fall into.

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	2017-01-31
2	2016-12-01	2017-01-24	87	2017-02-01	2017-02-28
2	2016-12-01	2017-01-24	87	2017-03-01	2017-03-31

```
CREATE TABLE cross_join AS
SELECT *
FROM Codeflix_Subscriptions
CROSS JOIN months
```

```
SELECT *
FROM cross_join
LIMIT 6;
```

2.5 Calculate Churn for Each Segment: Create status table with active and canceled columns

I will generate a table that shows whether or not the user was active or canceled in that particular month. I will add columns to count indicate which segment the user is in segment 37 or segment 80.

id	month	is_active_87	is_canceled_87	is_active_30	is_canceled_30
1	2017-01-01	1	0	0	0
1	2017-02-01	0	1	0	0
1	2017-03-01	0	0	0	0
2	2017-01-01	1	1	0	0
2	2017-02-01	0	0	0	0
2	2017-03-01	0	0	0	0

```
CREATE TABLE status_table 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=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_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=30 AND subscription_end BETWEEN
      first_day AND last_day THEN 1 ELSE 0
  END AS is_canceled_30
FROM cross_join;

SELECT *
FROM status_table
LIMIT 6;
```

2.6 Calculate Churn for Each Segment: Add a table to count the active users and cancellations in each segment

I will add a table that groups by month and sums the number of active and canceled users in each month and each segment.

Month	SumActive87	SumCanceled87	SumActive30	SumCanceled30
01	278	70	291	22
02	462	148	518	38
03	531	258	716	84

```
SELECT
  STRFTIME('%m', month) AS Month,
  SUM(is_active_87) AS SumActive87,
  SUM(is_canceled_87) AS SumCanceled87,
  SUM(is_active_30) AS sumActive30,
  SUM(is_canceled_30) AS SumCanceled30
FROM
  status_table
GROUP BY
  month
```

2.7 Calculate Churn for Each Segment: Modify SELECT statement

I will calculate the churn rate for each segment and round the result two digits after the comma.

Month	ChurnRate87	ChurnRate30
01	0.25	0.08
02	0.32	0.07
03	0.49	0.12

```
WITH temp_table AS
(
SELECT
    STRFTIME('%m', month) AS Month,
    SUM(is_active_87) AS SumActive87,
    SUM(is_canceled_87) AS SumCanceled87,
    SUM(is_active_30) AS sumActive30,
    SUM(is_canceled_30) AS SumCanceled30
FROM
    status_table
GROUP BY
    month
)
SELECT
    Month,
    ROUND((1.0 * SumCanceled87 / SumActive87), 2) AS
    ChurnRate87,
    ROUND((1.0 * SumCanceled30 / SumActive30), 2) AS
    ChurnRate30
FROM
    temp_table
```

3. Results & Discussion

3.1 Results and Discussion

Codeflix has been open for 4 months (Dec 1, 2016 - March 31, 2017). There is enough data to analyze churn for 3 of those months (January, February, March).

Marketers have divided customers into 2 segments: Segment 87 and Segment 30.

The overall churn rate for segment 87 has been steadily increasing whereas the churn rate for segment 30 has remained relatively flat. Segment 30 has a much lower churn rate (~75%) than Segment 87.

Therefore, Codeflix should focus on expanding the customers in Segment 30.

In addition, the churn rate of both segments increased from February to March. A separate analysis should be made about the month of March.

Month	Segment 87 Churn %	Segment 30 Churn %
January	25%	8%
February	32%	7%
March	%49	%12

**Thank you for your time
and attention.**