Calculating Churn Rates

Hassan A. Zeidan

Table of Contents

1. Get familiar with Codeflix
2. What is the overall churn rate by month?
3. Compare the churn rates between segments

1. Get familiar with Codeflix

1.1 Get familiar with Codeflix

Codeflix is a monthly video streaming service. Using the query on the right we can see the first 100 rows in the subscriptions table.

Note: There are four fields in that table.
- id
- subscription_start
- subscription_end
- segment

```
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
4	2016-12-01	2017-02-12	87

1.2 Get familiar with Codeflix

How many different segments do you see?

Based on the results in the table, there are two different segments. Using the Distinct function we can see that each segment has 1000 subscribers.

```
SELECT *
FROM subscriptions
LIMIT 100;

Select DISTINCT segment, count(*) as Segment_Count
 From subscriptions
 Group by segment;
```

Segment	Segment_Count
30	1000
87	1000

1.3 Get familiar with Codeflix

Determine the range of months of data provided. Which months will you be able to calculate churn for?

There are four month between the minimum subscription start and max subscription start. Given that subscribers can only cancel after 31 days then the month that we will be able to calculate churn for are the three months of January, February, & March.

```
SELECT Min(subscription_start),
 Max(subscription_start)
 From subscriptions;
```

Min(subscription_start)	Max(subscription_start)
2016-12-01	2017-03-30

2. What is the overall churn rate by month?

2.1 What is the overall churn rate by month?

What are Churn Rate?

A *churn rate* is the percent of subscribers to a monthly service who have canceled in any given month. Churn rates will help a company like Codeflix in determining how many people are leaving each month and why are they leaving based on each segment. The formula to calculate churn rates is:

$$\frac{cancellations}{total\ subscribers}$$

2.2 What is the overall churn rate by month?

What is the overall churn trend since the company started?

Before we can calculate the churn rate since the company started we needed to create a temporary table "**months**" for each month of January, February, and March. Then we cross joined the table with the subscriptions table. We named this table "**cross_join**".

```
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
  ),
```

2.3 What is the overall churn rate by month?

What is the overall churn trend since the company started?

Using the "cross_join" table we just created we create another temporary "status" table. Using the id from "cross_join" and then join the tables and used CASE WHEN to find any users from segment 87 who existed prior to the beginning of the month and used CASE WHEN again for users from segment 30 in order to calculate the number of active and cancelled users.

```
status AS (
  SELECT
    id,
    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
),
```

2.4 What is the overall churn rate by month?

What is the overall churn trend since the company started?

Lastly we aggregate the status table with the status_aggregate table to sum the sum the active and canceled users in segments 30 and 87 and group it by month to then use these results to find the churn rates. The result churn rates are below:

```
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
Group by month
) SELECT month,
1.0*sum_canceled_87/sum_active_87 as Churn_87,
1.0*sum_canceled_30/sum_active_30 AS Churn_30
From status_aggregate;
```

Month	Churn_87	Churn_30
2017-01-01	0.251798561151079	0.0756013745704467
2017-02-01	0.32034632034632	0.0733590733590734
2017-03-01	0.485875706214689	0.11731843575419

3. Compare the churn rates between segments

3.1 Compare the churn rates between segments

Which segment has a lower churn rate?

Looking at both the churn rates data in the table and the churn rates graph we can see that the Churn rate of segment 30 is much lower than churn of 87.   Codeflix should review their marketing strategy in segment_87 and try to find why there are more cancellations compared to segment_30.

Churn Rate Comparison

	January 1, 2017	February 1, 2017	March 1, 2017
Churn_87	0.251798561	0.32034632	0.485875706
Churn_30	0.075601375	0.073359073	0.117318436

Churn_87 Churn_30