By Michael Horwitz

codecademy

Capstone: Churn Rate

Learn SQL from Scratch

Background

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

Our objectives are:

1. Determine what is the overall churn rate by month
2. Compare the churn rates between segments

Outline of Presentation

1. Explore Codeflix Dataset
2. Determine Churn Rate
3. Compare Churn Rate between User Segments

1.Get Familiar with Codeflix

The objectives here are to investigate

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

1.1 How many months has the company been operating?

The companies first subscription in the dataset is 2016-12-01 and the last subscription is 2017-03-31. Therefore it seems they have been operating for 4 months

MIN(subscription_start)	MAX(subscription_start)	MIN(subscription_end)	MAX(subscription_end)
2016-12-01	2017-03-30	2017-01-01	2017-03-31

```
SELECT MIN(subscription_start),
MAX(subscription_start), MIN(subscription_end),
MAX(subscription_end)
FROM subscriptions;
```

1.2 Which months do you have enough information to calculate a churn rate?

The companies first subscription cancellation in the dataset is 2017-01-01 and the last subscription cancellation is 2017-03-31. The first subscriptions start in 2016-12-01.

Therefore using the formula *active users at the beginning of the month / total cancellations between the first and the last of the month* we can calculate churn rate for 3 months, January, February and March 2017.

We cannot calculate for December 2016 since there are no active users at the beginning of the month.

MIN(subscription_start)	MAX(subscription_start)	MIN(subscription_end)	MAX(subscription_end)
2016-12-01	2017-03-30	2017-01-01	2017-03-31

```
SELECT MIN(subscription_start),
MAX(subscription_start), MIN(subscription_end),
MAX(subscription_end)
FROM subscriptions;
```

1.3 What segments of users exist?

There are 2 segments of users. 87 and 31.

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

segment
87
30

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

The objective here is the determine the churn rate month over month. Before we do that we need to built a number of temporary tables to calculate churn for each month. Then we will query the tables to compare each months churn.

2.Temporary Tables

```
WITH months AS
 (SELECT '1' AS month, '2017-01-01'as first_day, '2017-01-31'as last_day
  UNION
  SELECT '2' AS month,'2017-02-01'as first_day, '2017-02-28'as last_day
  UNION
  SELECT '3' AS month,'2017-03-01'as first_day, '2017-03-31'as last_day
 ),

 cross_join AS
( SELECT *
  FROM months
  CROSS JOIN
 subscriptions
 ),

 status AS
 (SELECT id, first_day AS month, subscription_start, subscription_end,
 CASE
  WHEN (subscription_start < first_day) AND (segment = 87) THEN 1
   ELSE 0
  END is_active_87,
CASE
  WHEN (subscription_end BETWEEN first_day AND last_day) AND (segment = 87) THEN
1
   ELSE 0
  END is_cancelled_87,
CASE
 WHEN (subscription_start < first_day) AND (segment = 30) THEN 1
   ELSE 0
  END is_active_30,
CASE
 WHEN (subscription_end BETWEEN first_day AND last_day) AND (segment = 30) THEN
1
   ELSE 0
  END is_cancelled_30
 FROM cross_join
 ),
```

```
 status_aggregate_monthly AS (
 SELECT month, SUM(is_active_87) AS sum_active_87,
SUM(is_active_30) AS sum_active_30, SUM(is_cancelled_87) AS
sum_cancelled_87, SUM(is_cancelled_30) AS sum_cancelled_30
 FROM status
 GROUP BY month),

all_segment_monthly_churn AS (
SELECT month, ((1.0 * sum_cancelled_87 +sum_cancelled_30) /
(sum_active_87 + sum_active_30)) AS churn_rate_all_segments
FROM status_aggregate_monthly
)
```

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

The overall churn trend is BAD! The percentage of active users cancelling each month is growing month by month. 16% in January, 17% in February and 22% in March.

month	churn_rate_all_segments
2017-01-01	0.16140350877193
2017-02-01	0.172701949860724
2017-03-01	0.222511385816526

```
SELECT *
FROM all_segment_monthly_churn;
```

3. Compare the churn rates between user segments

The objective here is the determine which segment of users should the company focus on expanding. To do this we need to compare churn rates between our 2 user segments.

We will build another temporary table that looks at monthly churn broken down by segment.

3. Compare the churn rates between user segments

Based on the data it is clear user segment 30 has a lower churn rate and therefore the company should focus on them.

month	churn_87	churn_30
2017-01-01	0.25089605734767	0.0756013745704467
2017-02-01	0.2756052141527	0.0703703703703704
2017-03-01	0.339920948616601	0.107969151670951

```
monthly_churn_by_segment AS (
SELECT month, sum_active_87 AS active_users_87,
sum_cancelled_87 AS cancellations_87, (1.0 *
sum_cancelled_87 / sum_active_87) AS churn_87,
sum_active_30 AS active_users_30, sum_cancelled_30 AS
cancellations_30, (1.0 * sum_cancelled_30 /
sum_active_30) AS churn_30
 FROM status_aggregate_monthly)

 SELECT month, churn_87, churn_30
 FROM monthly_churn_by_segment
 ORDER BY month ASC;
```