Churn Rates with Codeflix

Learn SQL from Scratch

Date: 8th July 2018, Student: Dave Tuchan

Table of contents

1. Get familiar with the company.

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

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

3. Compare the churn rates between user segments.

- Which segment of users should the company focus on expanding?

1. Get familiar with the company

Q1: How many months has the company been operating?

A: Codeflix started his operations the first of December 2016. The company is operating for four months now.

Q2: Which months do you have enough information to calculate a churn rate?

A: Codeflix requires a minimum subscription length of 31 days, which means that the churn rate can be calculated 31 days after the first subscription. We can calculate the churn rate in January, February and March 2017.

Q3: What segments of users exist?

A: There are two type of users, groups by 30 & 87.

```sql
--Q1 Operating months
--Q2 Churn rate months
SELECT MIN(subscription_start) AS 'First subscription',
       MAX(subscription_start) AS 'Last subscription'
FROM subscriptions;

--Q3 User segments
SELECT DISTINCT Segment
FROM subscriptions;
```

Query Results	
First subscription	Last subscription
2016-12-01	2017-03-30

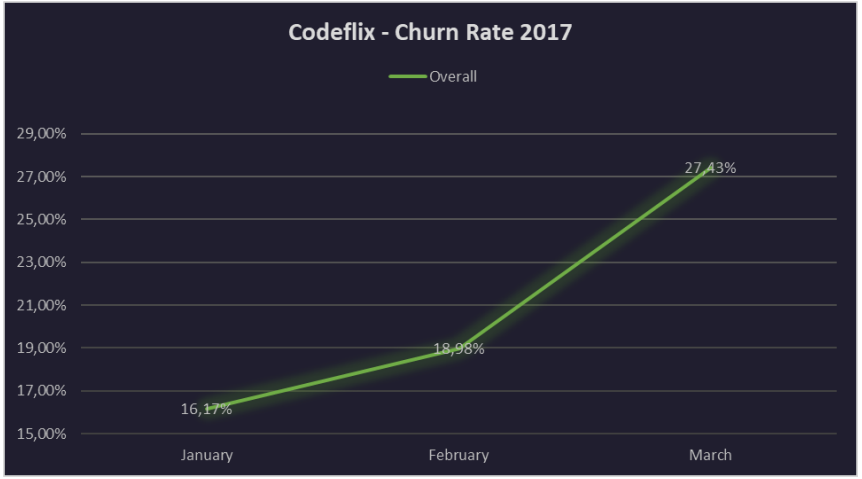
segment
87
30

2. Overall churn trend since the company started

Q1: What is the overall churn trend since the company started?

A: The company started in December 2016, because of subscription policy. There is no churn rate calculation available for this month.

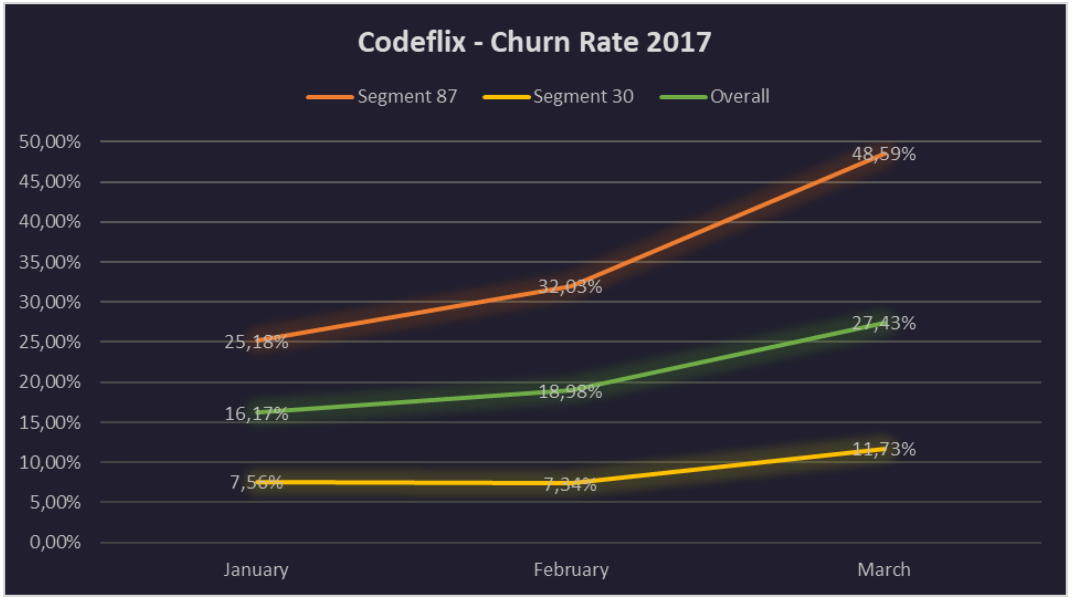
As we can see there is a positive churn rate growth, which means that the percentage of canceled subscriptions compared to the active subscriptions is growing each month.



3. Compare the churn rates between user segments

Q1: Which segment of users should the company focus on expanding?

A: The company should focus on investigating the causes of the leaving subscribers in segment 87. Based on the investigated results, they can start campaigns and expand this segment. If this will take no effect, they need to think about another strategy, like searching for a new segment.



Compare / Overall churn rate (code 1/3)

```sql
--Temporary table: Months
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

--Temporary table: Subscriptions combined with Months
cross_join AS (
SELECT *
FROM subscriptions
CROSS JOIN months
),
```

Compare / Overall churn rate (code 2/3)

```
--Temporary table: Active / Canceled users
status 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 = '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 = '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_end
               BETWEEN first_day
               AND last_day)
          THEN 1
         ELSE 0
       END AS 'is_canceled_30'
FROM cross_join
),
```

Compare / Overall churn rate (code 3/3)

```sql
--Temporary table: Aggregated fields
status_aggregate AS (
SELECT month,
       SUM(is_active_30) AS 'sum_active_30',
       SUM(is_active_87) AS 'sum_active_87',
       SUM(is_canceled_30) AS 'sum_canceled_30',
       SUM(is_canceled_87) AS 'sum_canceled_87',
       SUM(is_active_30) +
       SUM(is_active_87) AS 'Sum_active_total',
       SUM(is_canceled_30) +
       SUM(is_canceled_87) AS 'Sum_canceled_total'
FROM status
GROUP BY 1
ORDER BY 1 ASC
)
```

```sql
--Calculate churn rate statistics
SELECT month,
       ROUND(1.0 * sum_canceled_87 /
             sum_active_87 * 100, 2) AS 'Segment 87',
       ROUND(1.0 * sum_canceled_30 /
             sum_active_30 * 100, 2) AS 'Segment 30',
       ROUND(1.0 *  Sum_canceled_total /
             Sum_active_total * 100, 2) AS 'Overall'
FROM status_aggregate
GROUP BY 1
ORDER BY 1 ASC;
```

Date: 8th July 2018, Student: Dave Tuchan