codecademy

Codeflix Overview

Focus: Churn Rate

Table of Contents

1. Company Overview
2. Churn rate trends
3. Churn rates between segments

Company Overview

1.1 Company Overview

- Codeflix has been operating for 4 months
- First customer started their subscription in 12-1-2016
- 2,000 subscriptions since the start of the company
- Data collected:
- Id
- Subscription start
- Subscription end
- Segment

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	30
4	2016-12-01	2017-02-12	30

Churn Rate Trends

2.1 Churn Rate Trends

- Churn rate has been increasing month over month since January
- Calculated churn by adding all active users for each segment, and adding all cancelled users by segment
- Then dividing the cancelled users against the active users

Month	Segment 87	Segment 30
January	25.2%	7.6%
February	32.0%	7.3%
March	48.6%	11.7%

```
status AS (
  SELECT
    id,
    segment,
    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,
             segment,
    SUM(is_active) AS sum_active,
    SUM(is_canceled) AS sum_canceled
  FROM status
  GROUP BY month,
             segment)
SELECT month,segment,
1.0 * sum_canceled / sum_active AS churn_rate
FROM status_aggregate
GROUP BY month,segment
```

Churn Rates Between Segments

3.1 Churn Rate Between Segments

- Was able to calculate churn by grouping months and segments
- Removed lines that manually aggregate active subscriptions and cancellations
- This will allow automation of calculating churn by segments, when new segments are added moving forward

```
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
    id,
    segment,
    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,segment,
    SUM(is_active) AS sum_active,
    SUM(is_canceled) AS sum_canceled
  FROM status
  GROUP BY month,segment)
SELECT month,segment,
1.0 * sum_canceled / sum_active AS churn_rate
FROM status_aggregate
GROUP BY month,segment

;
```

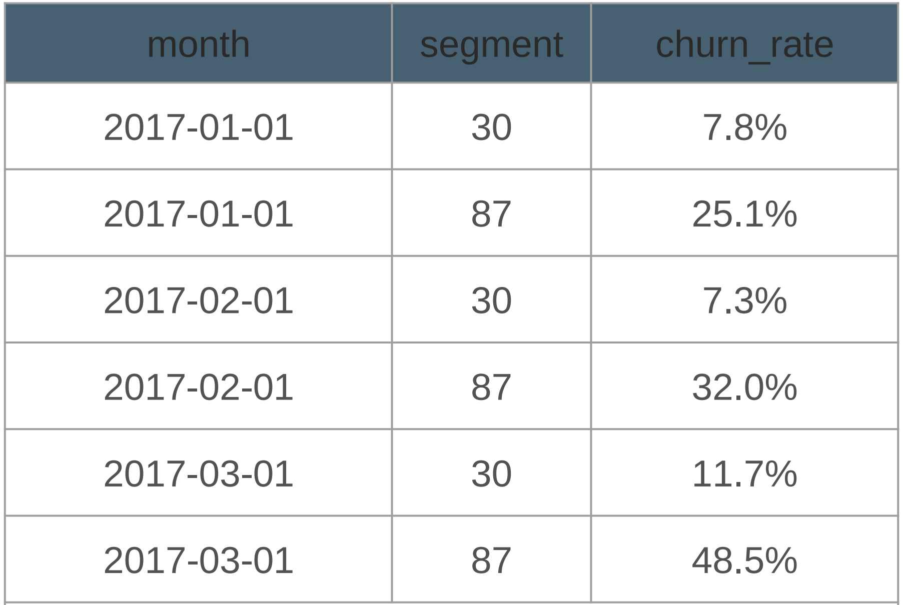
3.2 Churn Rates Between Segments

- The image to the right shows the results of the query written on the previous slide
- As you can see this groups by segments, and as new segments are added – will include them as well

- Segment 30 has significantly lower churn than segment 87

- Segment 30 also has higher consistency in churn

- Segment 87 has been increasing at a much higher rate, topping at 48.5% in March 2017, compared to only 11.7% for segment 30



month	segment	churn_rate
2017-01-01	30	7.8%
2017-01-01	87	25.1%
2017-02-01	30	7.3%
2017-02-01	87	32.0%
2017-03-01	30	11.7%
2017-03-01	87	48.5%

*Results of automating churn rate by segment

Recommendations

- Analyze the success of segment 30 in their lower churn and try to replicate
- Dive into segment 87, and figure out why so many customers are cancelling