codecademy

Idriss Bamba

Calculating Codeflix's Churn Rates Project
October 15, 2018
Learn SQL from Scratch Capstone Project

CODEFLIX TABLES of Contents

CODEFLIX TABLES of Contents

1. **How many months has the company been operating?**

CODEFLIX company have been operating for 4 months.

From 2016-12-01 -2017-03-31

CODEFLIX TABLES of Contents

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

We only have enough information for 3 months to calculate a churn rate. Codeflix requires a minimum subscription length of 31 days, so a user can never start and end their subscription in the same month.

CODEFLIX TABLES of Contents

3. What segments of users exist?

Segment (30)and (87) identify which segment the subscription owner belongs to.

CODEFLIX TABLES of Contents

4. What is the overall churn rate by month?

Query Results

month	churn_rate_87	churn_rate_30
2017-01-01	0.251798561151079	0.0756013745704467
2017-02-01	0.32034632034632	0.0733590733590734
2017-03-01	0.485875706214689	0.11731843575419

CODEFLIX TABLES of Contents

5. Compare the churn rates between segments

The Churn_rate_30 segment perform better . As of 2017-03-01 it was 0.11731843575419 which is less to the Churn_rate_87 of 0.485875706214689 . The analysis tells us that less people are canceling their subscription in Churn_rate_30 compare to Churn_rate_87 . Therefore the CODEFLIX company is better of making more income with Churn_rate_30.

Query Results

month	churn_rate_87	churn_rate_30
2017-01-01	0.251798561151079	0.0756013745704467
2017-02-01	0.32034632034632	0.0733590733590734
2017-03-01	0.485875706214689	0.11731843575419

CODEFLIX TABLES of Contents

6. Which segment of users should the company focus on expanding?

Churn_rate_30 should be the focus for CODEFLIX should focus on expanding Churn_rate_30 is as a lower number of subscription that get canceled compare to Churn_rate_87

Query Results

month	churn_rate_87	churn_rate_30
2017-01-01	0.251798561151079	0.0756013745704467
2017-02-01	0.32034632034632	0.0733590733590734
2017-03-01	0.485875706214689	0.11731843575419

My Capstone SQLCodes

```
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,
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
) ,
status_aggregate AS
 (  SELECT
month,
    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_rate_87,
  1.0 * sum_canceled_30/sum_active_30 AS
churn_rate_30
FROM status_aggregate;
```