codecademy

Exploring Codeflix User Churn Rate

David Akers

Table of Contents

1. An overview of the Codeflix dataset
2. Codeflix monthly churn rates
3. Comparing the churn rates

Section 1: An overview of the Codeflix dataset

1.1 Data elements

We begin by exploring the dataset. The columns are an ID field, subscription start and end dates, and the customer segment. We will explore the ranges of the subscription start and end dates in the next slide, but note here that the segment field takes values ‘87’ and ‘30’ in this data set.

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 Months of data availability

We note that the Codeflix data contains start and end dates for these subscriptions. From the table below, we see that the start dates range from December 1st, 2016 to March 30th, 2017. The end dates range from January 1st, 2017 to March 3rd, 2017. Thus, we will be able to calculate churn rates for the first 3 months of 2017. We cannot calculate a churn rate for December 2016, because there were no subscriptions that ended in that month.

Activity	First date	Last date
Start	2016-12-01	2017-03-30
End	2017-01-01	2017-03-31

```
SELECT MIN(subscription_start) AS range_start,
MAX(subscription_start) AS range_end
FROM subscriptions;

SELECT MIN(subscription_end) AS range_start,
MAX(subscription_end) AS range_end
FROM subscriptions;
```

Section 2: Codeflix monthly churn rates

2.1 Overall and monthly churn rates

We can use the subscription starts and cancellations to calculate an overall churn rate. There were 2,000 subscriptions and 620 cancellations for a churn rate of .31 or 31% overall. The monthly churn rates are listed below. It is worth noting here that the churn rate was noticeably higher in March than the first two months. Codeflix may wish to explore this month.

Activity	First date
January	0.162
February	0.190
March	0.274

```
SELECT 1.0 *
(
  SELECT COUNT(*)
  FROM subscriptions
  WHERE subscription_end IS NOT NULL
 ) / (
  SELECT COUNT(*)
  FROM subscriptions
)
AS result,
(
  SELECT COUNT(*)
  FROM subscriptions
  WHERE subscription_end IS NOT NULL
 ) AS cancellations,
 (
  SELECT COUNT(*)
  FROM subscriptions
) AS subscriptions
 ;

*Monthly code too big to fit here
```

Section 3: Comparing churn rates

3.1 Churn rates for the two segments

We now wish to determine which segment of customers could benefit us the most by expanding. We will do this by calculating the monthly churn rates for the two segments. Note that all three months are listed below.

Month	Segment 87 churn rate	Segment 30 churn rate
2017-01-01	0.251	0.076
2017-02-01	0.320	0.073
2017-03-01	0.486	0.117

```
*Code to create the months, cross_join, and status
tables left off to conserve space.

---------------------------
status_aggregate AS (
  SELECT
    month,
    SUM(is_active_87) AS sum_active_87,
    SUM(is_canceled_87) AS sum_canceled_87,
    SUM(is_active_30) AS sum_active_30,
    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;
```

3.2 Conclusions

We calculated the churn rates for the two segments by month to ensure that we did not miss any interaction between time and segment. However, the segment 30 churn rate was noticeably lower for every month then the segment 87 churn rate. Thus, we can conclude that the most reasonable segment to expand is segment 30.

Month	Segment 87 churn rate	Segment 30 churn rate
2017-01-01	0.251	0.076
2017-02-01	0.320	0.073
2017-03-01	0.486	0.117

THANKS!