codecademy

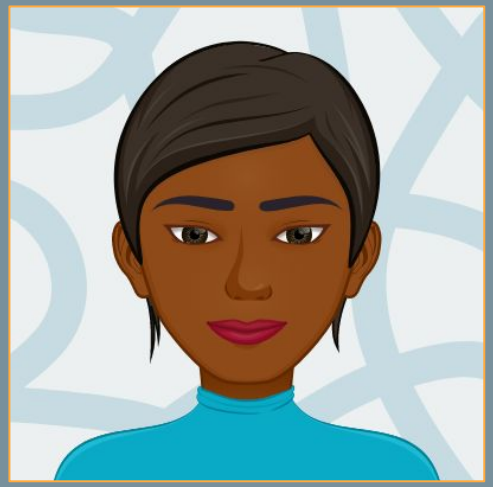
Which Group to Choose? Codeflix

Learn SQL from Scratch

Presented by Dijan Matheson

England - 11th September, 2018

Welcome!



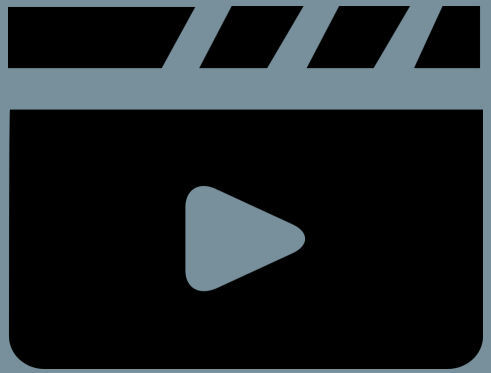
About Presenter

Dijan Matheson- Data Specialist from England

Interests- Coding, hiking and travelling

Contents of Presentation

1. What we know about Codeflix
2. Overall Churn Trends
3. Comparing Codeflix User Groups
4. Which Codeflix User Group to Choose?



1.0 What we know about Codeflix

Codeflix is a streaming video startup

- Codeflix has been in operation for 4 months [December ‘16 - March ‘17].
- We have a table called subscriptions with the following columns:- id, subscription_start date / end date and segment.
- To calculate how many months Codeflix has been in operation. I used a SELECT query to determine the first and last subscription start dates. This generates the range of dates in which users subscribed to Codeflix.

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
5	2016-12-01	2017-03-09	87
6	2016-12-01	2017-01-19	87

```
11
12  -- calculate the range of months
13  --check 1 enough data during first month for total subscriptions
14  --check 2 enough data during last month for cancellations by end of month
15  SELECT MIN(subscription_start), MAX(subscription_start),
16         MIN(subscription_end), MAX(subscription_end)
17  FROM subscriptions
18  ;
19
```

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

1.0 What we know about Codeflix- Continued

Churn Rate measures the percentage of subscribers who cancel a subscription to a service in a given time

- Based on the available data, we can calculate the Churn Rates for January ‘17 - March ‘17.
- The minimum subscription duration to Codeflix is 31 days and users did not have the opportunity to cancel their subscriptions during the month of December ‘16. The Churn calculation takes into account cancellations during the month, meaning that the Churn Rate calculation cannot be performed for December ‘16.

CHURN RATE = CANCELLATIONS / TOTAL SUBSCRIBERS

1.0 What we know about Codeflix-Continued

- We have two segments of users: 30 and 87
- This was calculated using the SELECT DISTINCT query, to produce a table that identifies each type of segment in the subscriptions table.

```
6
7   --discover the different segments of users
8   SELECT DISTINCT segment
9   FROM subscriptions
10  ;
11
```

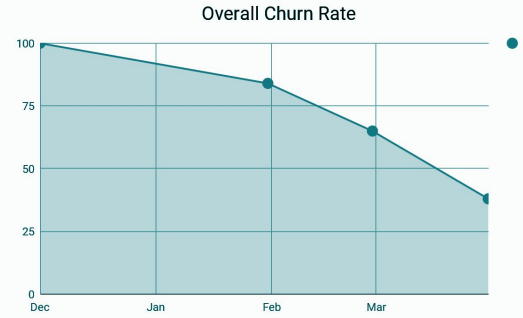
Out of curiosity I included the COUNT () query to calculate how many users were in each segment. This gave me a better feel for the magnitude of the users in the dataset

segment	Total_users
30	1000
segment	Total_users
87	1000

```
20  --count number of users in each section
21  SELECT segment, COUNT (*) as Total_users
22  FROM subscriptions
23  WHERE segment = 30
24  ;
25  SELECT segment, COUNT (*) as Total_users
26  FROM subscriptions
27  WHERE segment = 87
28  ;
29
```

2.0 Codeflix Overall Churn Trends

- The overall Churn Rate for Codeflix to date is 0.63
- This means that 63% of subscribers to Codeflix have cancelled their subscriptions to date
- In order for Codeflix to remain in existence, the percentage intake of new subscribers must exceed the percentage Churn rate




```sql
--overall churn rate at Codeflix
SELECT ROUND(1.0 *
( SELECT COUNT(*)
  FROM subscriptions
  WHERE subscription_start < '2017-01-01'
  AND (subscription_end
    BETWEEN '2017-01-01'
    AND '2017-03-31'
  )) / (
SELECT COUNT(*)
FROM subscriptions
WHERE subscription_start < '2017-01-01'
AND ((subscription_end >= '2017-01-01')
  OR (subscription_end IS NULL)
)),2) AS Overall_Churn_Rate
;
```

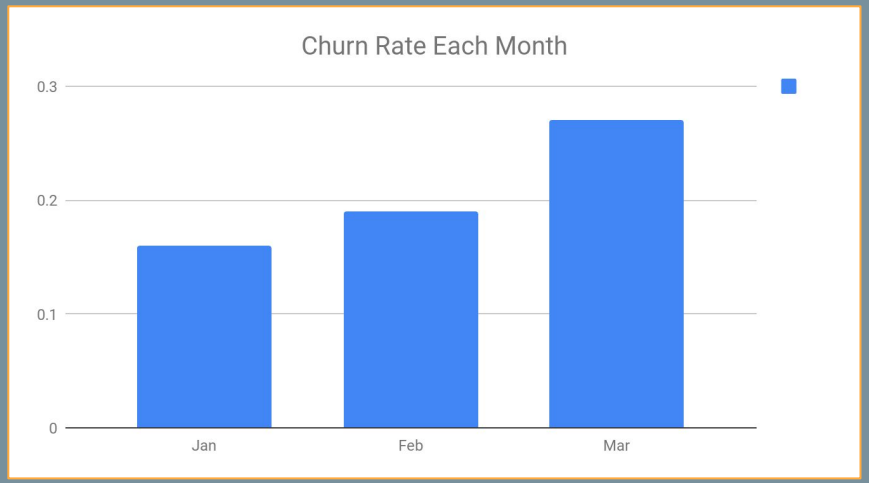
To calculate the overall churn rate for the dataset I did the following;

Numerator - Counts the number of subscribers that have cancelled. The Aggregate function COUNT () was used to count the number of rows that met a given set of conditions. A WHERE query was used, to identify subscribers that were subscribed on 1st January but cancelled on or before 31st March.

Denominator - Counts the total number of subscribers on 1st January. The same queries were used as above but the conditions of the WHERE query were change.

The ROUND query was included to tidy the result and the AS query was used to rename the column.

2.0 Codeflix Overall Churn Trends- Continued



Month	Churn_Rate_Each_Month
Jan	0.16
Feb	0.19
Mar	0.27

The Churn Rate has progressively increased each month since Codeflix started.

With March ‘17, seeing 27% of total subscribers cancelling their subscriptions.

3.0 Comparing Codeflix User Groups

```
47  --calculate churn rate each month
48  WITH months AS (
49  SELECT
50    '2017-01-01' as first_day,
51    '2017-01-31' as last_day
52   UNION
53   SELECT
54    '2017-02-01' as first_day,
55    '2017-02-28' as last_day
56   UNION
57   SELECT
58    '2017-03-01' as first_day,
59    '2017-03-31' as last_day
60    ),
61  cross_join AS (
62   SELECT *
63   FROM subscriptions
64   CROSS JOIN months
65   ),
```

Calculating the churn rate for each segment

Step 1. To know the first and last day of the months for each user in the subscriptions table. A temporary months table was created and then CROSS JOINED with the subscriptions table. To combine all rows of months with all rows of subscriptions.

Step 2. The temporary cross_join table was used to create a status table with 4 categories - segment 30 and 87 users that were active/ had cancelled. The status table represents the active users on the first day of the month and the users that had cancelled by the last day of the month for each segment.

Continued

```
122  status AS (
123  SELECT
124    id,
125    first_day as month,
126    CASE
127      WHEN (segment = 87)
128           AND
129           (subscription_start < first_day)
130           AND
131           (subscription_end >= first_day
132           OR subscription_end IS NULL)
133                 THEN 1
134                 ELSE 0
135                 END as is_active_87
136    ,
137    CASE
138       WHEN (segment = 30)
139            AND
140            (subscription_start < first_day)
```

3.0 Comparing Codeflix User Groups- Continued

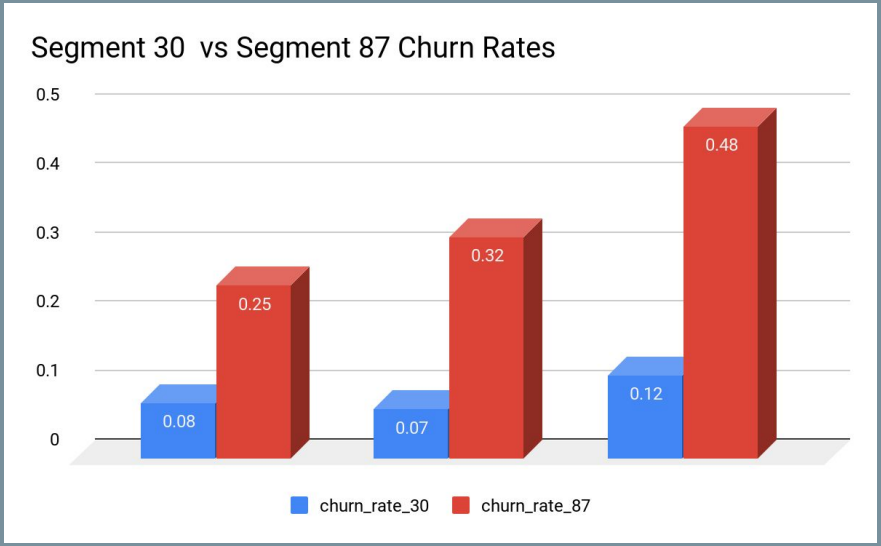
```
165  ,
166    status_aggregate AS (
167    SELECT
168    month,
169    SUM(is_active_87) as sum_active_87,
170    SUM(is_active_30) as sum_active_30,
171    SUM(is_canceled_87) as sum_canceled_87,
172    SUM(is_canceled_30) as sum_canceled_30
173    FROM status
174    GROUP BY month)
175
176   SELECT
177    month,
178    ROUND(1.0 * sum_canceled_87 / sum_active_87,2)
179    as churn_rate_87,
180    ROUND(1.0 * sum_canceled_30 / sum_active_30,2)
181    as churn_rate_30
182    FROM status_aggregate
183  ;
```

Step 3. The status_aggregate table is created. We use SUM to bring together users in each of the 4 categories for each month.

Step 4. We calculate the Churn Rate for each segment for each month

month	churn_rate_30	churn_rate_87
Jan	0.08	0.25
Feb	0.07	0.32
Mar	0.12	0.48

3.0 Comparing Codeflix User Groups- Continued



month	churn_rate_30	churn_rate_87
Jan	0.08	0.25
Feb	0.07	0.32
Mar	0.12	0.48

Comparison

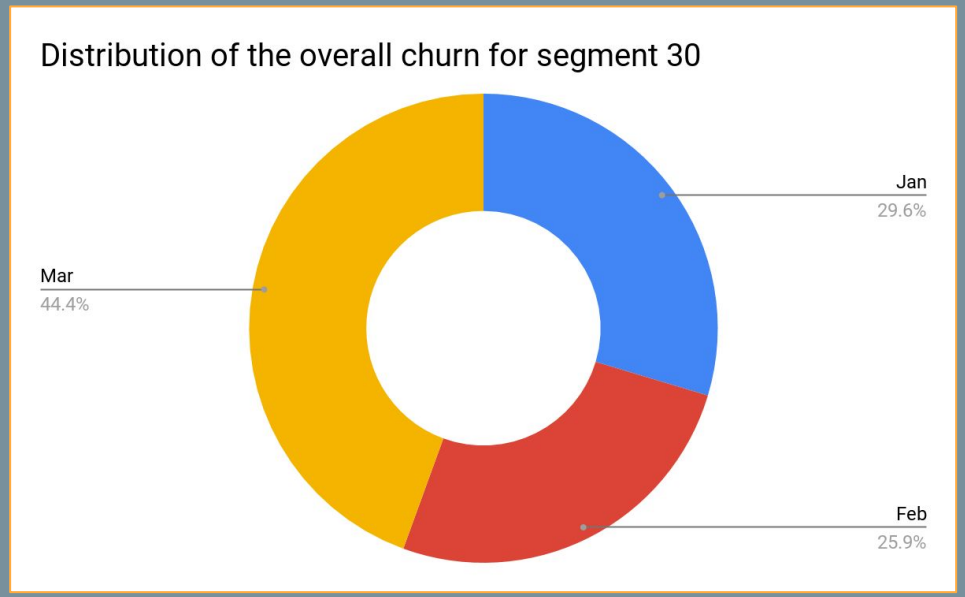
- Segment 30 has a lower overall churn rate than Segment 87
- Churn rates increased each month for segment 87 users
- The Churn rate for segment 30 users dropped slightly from January to February and then increased significantly from February to March

4.0 Which is the better Codeflix User Group?

Recommendations

- Based on the data my suggestion would be to focus on expanding segment 30, because so far it's easier to retain users in segment 30
- We need some feedback from users in segment 87, to understand why churn rates are so high
- Radical changes need to be implemented and Churn rates need to be monitored going forward

4.0 Which is the better Codeflix User Group?-Continued



Segment 30

Retention in February was very good

What happened in February? Why were subscribers less likely to cancel their subscriptions?

Dijan Matheson
Github: @ dijana181
dijanmatheson@gmail.com