codecademy

Capstone: Churn Rates Codeflix

Learn SQL from Scratch

Troy Moore
September 4, 2018

TABLE OF CONTENTS

Table of Contents

1. Getting familiar with Codeflix

1.1 Getting familiar with Codeflix

Codeflix is a monthly video streaming service

To get familiar with the tables schema
we selected all columns in the table and limited the data to 100 rows.

```
SELECT *
FROM subscriptions
LIMIT 100;
```

Database Schema	
subscriptions	
id	INTEGER
subscription_start	TEXT
subscription_end	TEXT
segment	INTEGER

Id	subscription_start	subscription_end	segment
1	2016-12-01	2017-02-01	87
2	2016-12-01	2017-01-24	87

1.2 Getting familiar with Codeflix

- What segments of users exist?
- Using SELECT DISTINCT we can see the codeflix data contains 2000 customers
- Separated in two segments 87 & 30 with 1000 customers each.

```
SELECT DISTINCT segment,
COUNT(*) as Number of customers
FROM subscriptions;
```

- How many months has codeflix been operating?

Min_start_date	max_start_date
2016-12-01	2017-03-30

```
SELECT MIN(subscription_start) AS
'min_start_date',
MAX(subscription_start) AS
'max_start-date'
FROM subscriptions;
```

- *We only can compute the churn rate for 3 months because December 2016 has no end dates*

2. Churn rates

Churn = Cancellations / Total subscribers

2.1 Churn rates

What is the overall churn tendency since the company started?

To figure this we

- Create temporary table for MONTHS
- Then CROSS JOIN the months table with subscriptions table
- Then we compute the active and cancelled users from the joined tables using a CASE statement.

```
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 )
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),
```

2.2 Churn rates

What is the overall churn tendency since the company started?

Finally aggregate the number of cancelled and active members, then sort by date. As shown the respective monthly churns are 16,19, and 27% for Jan, Feb, and Mar. (Simplified)

Month	Overall churn_rate
2017-01-01	0.161687170474517
2017-02-01	0.189795918367347
2017-03-01	0.274258219727346

Month Simplified	Churn Rate
JAN	16%
FEB	19%
MAR	27%

```
status_aggregate AS
(SELECT
 month,
 SUM(is_active) as sum_active,
 SUM(is_canceled) as sum_canceled
FROM status
GROUP BY month)
SELECT month,
1.0*sum_canceled/sum_active as 'overall
churn_rate'
FROM status_aggregate;
```

2.3 Comparison of churn rate

We compare the churn rates between user segments

To compare segment 87 & 30 we add an AND statement to the status table query.

Segment 87

```
status AS
(SELECT id, first_day as month,
...
-- the is active 87 segment
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,
-- the is cancelled 87 segment
CASE
WHEN (subscription_end BETWEEN first_day AND
last_day) AND ( segment= 87)
THEN 1
ELSE 0
END as is_canceled_87,
…
FROM cross_join),
```

Segment 30

```
status AS
(SELECT id, first_day as month,
...
-- the is active 30 segment
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,
-- the is cancelled 30 segment
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),
```

2.3 Comparison of churn rate

The last step to compare the churn rates between user segments

We must modify the status_aggregate table to give us the churn rate for segment 87 and 30 along with a total churn rate.

```
status_aggregate AS
(SELECT
month,
SUM(is_active) as sum_active,
  SUM(is_active_30) as sum_active_30,
  SUM(is_active_87) as sum_active_87,
SUM(is_canceled) as sum_canceled,
  SUM(is_canceled_30) as sum_canceled_30,
  SUM(is_canceled_87) as sum_canceled_87
FROM status
GROUP BY month)
-- computes the churn rate total
SELECT month,
1.0*sum_canceled/sum_active as
total_churn_rate,
1.0*sum_canceled_30/sum_active_30 as
churn_rate_30,
1.0*sum_canceled_87/sum_active_87 as
churn_rate_87
FROM status_aggregate;
```

Month	Total_churn_rate	Churn_rate_30	Churn_rate_87
2017-01-01	0.161687170474517	0.07560137457044	0.25179856115107
2017-02-01	0.189795918367347	0.07335907335907	0.32034632034632
2017-03-01	0.274258219727346	0.11731843575419	0.48587570621468

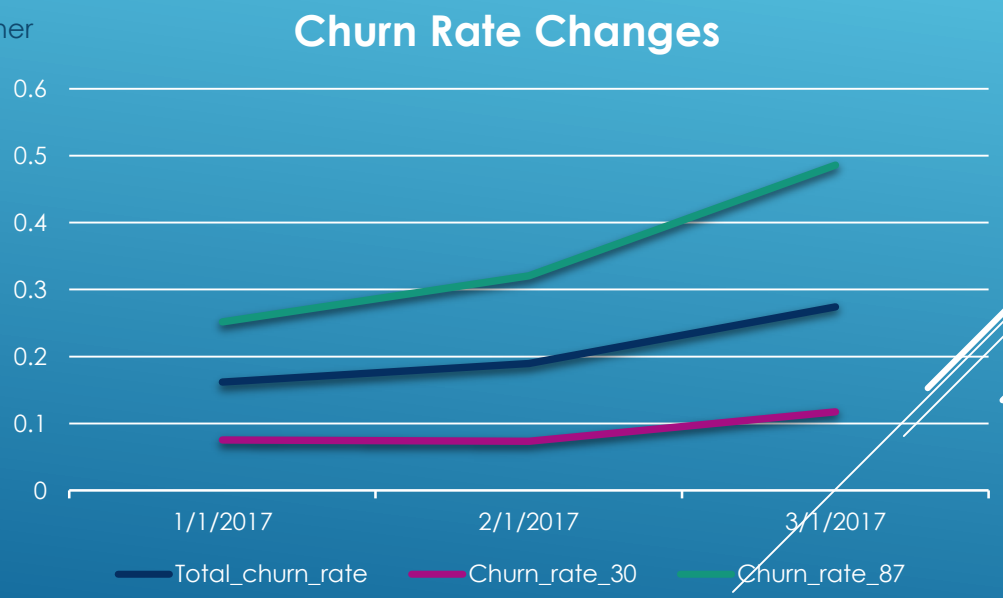
4. Conclusions

4.1 Conclusion

Analyzing data

Plotting the churn rates for all segments, segment 87 and segment 30, we observe an increasing churn rate pattern across all customer segments.

The churn rate for Segment 87 was consistently higher than for Segment 30.

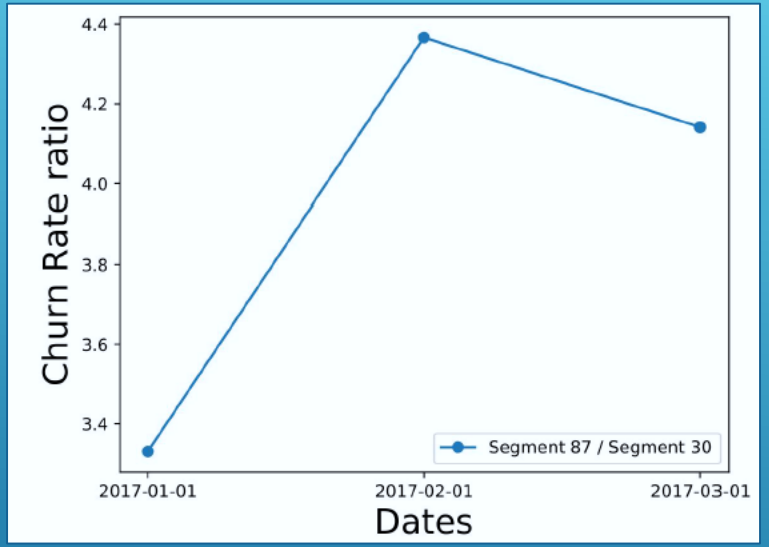


4.2 Conclusion

Analyzing data

Taking the ratio of the churn rates of segment 87 and 30, we observe that there is a factor of approximately 3-4 between these two segments.

The churn rate ratio is highest for the Month of February, where it reaches a value of 4.4



4.3 Concluding recommendations

Based on our findings we can see that the churn rate increased from Jan through March. Codeflix should identify what caused the increase in churn rates for these months and take actions to decrease the churn rate.

We also noticed that customer segment 87, had a very high churn rate in comparison to segment 30. We feel Codeflix should make changes to how they manage and market segment 87, in order to reduce the churn rate. Because users seem like likely to stay subscribed with segment 87.

Codeflix should work on expanding segment 30 and changing their approach to segment 87. That would reduce the overall churn rate. Lastly, Codeflix could benefit from added customer segments to test and implement different methods of reducing churn rates.