codecademy

Churn Analysis Capstone

Learn SQL from Scratch

Mackenzie Cesar

July 2018

Table of Contents

1. Getting Familiar w/ Data

1.1 Getting Familiar with Codeflix

- To begin the analysis of the Codeflix database we first need to get a feel for how the data is structured.
    - We can do this by selecting all the column in the database using * and limiting the number of rows that we see with the LIMIT clause.
    - Next we want to know how many different segments are included in the database. For this we use a DISTINCT clause to show the unique segments.
    - Finally, we need to know how many months of data we have to work with. For this we can simply use the MIN/MAX functions on the data field of interest.
    - The code and their outputs are shown in the adjacent figures.

```sql
-- Looking how the database is structured
SELECT *
 FROM subscriptions
 LIMIT 5;
--Looking for the unqiue segments
 SELECT DISTINCT segment
 FROM subscriptions;
 --Finding the date range for this field
 SELECT  MIN(subscription_start), MAX(subscription_start)
 FROM subscriptions;

```

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

segment
87
30

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

2.1 What is the overall churn rate?

To calculate the churn rate for the different months of data a few things need to be taken care of beforehand.

- Using the WITH clause, we can create a date tableau that will be later joined to the Codeflix data source. We join the tables using a CROSS JOIN.
- To find the active customers for the month, we have to find those that started their subscription before the month starts and didn't canceled their plan during the month or didn't cancel at all (NULL).
- To count the churn customers, we need to find the customer who canceled their plan anytime between the first day of the month and the last day of the month.
- Finally we create another table that aggregates the active and churned customer by the individual months. Now to calculate churn we need to multiply by 1.0 to force the number to float.

Month	Active Customers	Canceled Customers	Churn Rate
January 2017	569	92	16.2%
February 2017	980	186	18.9%
March 2017	1247	342	27.4%

```sql
--Question 2
--Creating temporary date table
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
),
--Using cross join to combine the temp. table to the Codeflix Source
cross_join AS(
SELECT *
FROM subscriptions
CROSS JOIN months),
--Using newly created table to find customers who have churned
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
   ),
  --Table to aggregate the results from the previous query
status_aggregate AS(
SELECT month,
  sum(is_active) as sum_is_active,
  sum(is_canceled) as sum_is_canceled
  FROM status
GROUP BY month)
--Finally the overall churn rate is found
SELECT month,1.0 * sum_is_canceled/sum_is_active as 'overall_churn'
FROM status_aggregate;
```

2.2 What is the churn rate by segment?

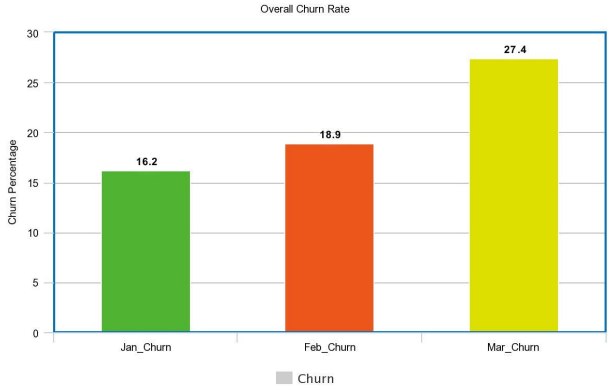
To calculate the churn rate for the different months of data and for the different segments we need to at at few more lines of code to the script from slide 2.1.

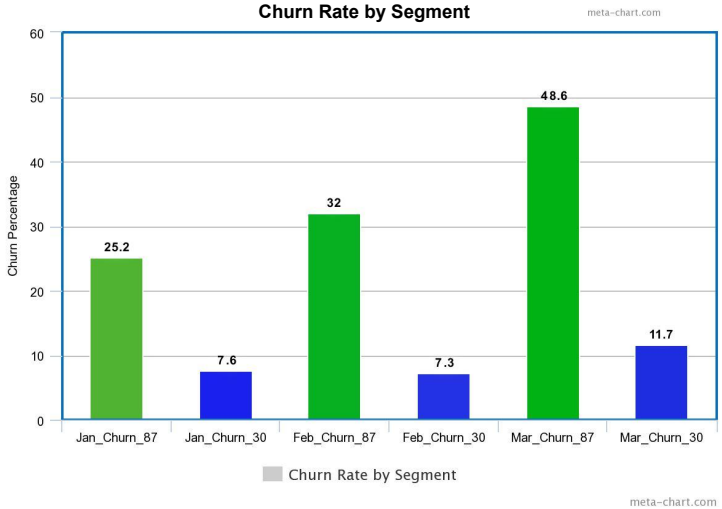
- To find the churn rate for the different segments, we simply need to add an AND operator to CASE statement differentiating the segments. And have CASE statements that aggregate the active/canceled members separately.
- The output of these adjustments are listed below

Month	Active_87	Active_30	Canceled_87	Canceled_30	Churn_87	Churn_30
Jan	278	291	70	22	25.2%	7.6%
Feb	462	518	148	38	32.0%	7.3%
Mar	531	718	258	81	48.6%	11.7%

```
status AS(
SELECT id, first_day AS month,
  CASE
   WHEN (subscription_start < first_day)
  AND
  segment = '87'
   AND (
     subscription_end > first_day
   OR subscription_end IS NULL) THEN 1 ELSE 0
  END as is_active_87,
CASE
   WHEN (subscription_start < first_day)
  AND
  segment = '30'
   AND (
     subscription_end > first_day
   OR subscription_end IS NULL) 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
),
```

2.2 What is the churn rate by segment?(cont'd)





Overall from January 2017 to March 2017, the subscriber churn rate has been increasing for Codeflix.

- For Jan to Feb, the overall churn rate increased 16%. From Feb to Mar, the churn increased by 45% with 27.4% of subscribers churn.

Looking at the individual segments helps us to see what segments are churning more than others.

- The churn rate in segment 87 increased by 27% going into Feb and then increased by 52% going into Mar. Both of the increases surpassed the increases for overall rates which shows that his segment isn't performing as well at the whole. They are also higher then the overall values
- The churn rate in segment 30 actually decreased by about 4% going into Feb. In the month of March this segment's churn rate increase by 60%, but it increase to 11.7% which is well below the overall average churn rate.

2. Conclusion

Where should Codeflix expand?

After looking at the result of the query I believe Codeflix should look into expanding the Segment 30. And here's why:

- Segment 30 had more active users, and was experiencing more month over month growth in active user then Segment 87.
- Segment 30 had a lower rate of churn then Segment 87, and was performing better than the overall churn rate.