

Giving weightage to last 30 day activity in approval systems

Author: [Vivek Pandey](#)

Date: May 2025

Summary

This document proposes two changes that our approval systems:

1. If we give a line to the user on date D, then we will reevaluate the user on D + 30 if they have not transacted by then. If the user no longer fits our approval criterion, the line given to them will be backscrubbed. However, if they first got a line ≥ 180 days ago, then their line will not be backscrubbed.
2. We will have an additional criterion at the time of approval - the user needs to be seen on our network for at least 3 days in the last 30 days for them to avail a line.

Context

Early May, [Shanker Mohan](#) made the observation that if we take all users acquired in March, then amongst these users, those who were seen for very few days in Feb had high delinquency and low retention. He argued therefore, that we should not acquire such users.

Why did we not think about this earlier? We had an unstated assumption - once we give a line to a user, we should not take it away except in unusual circumstances (like we did in May 2024). So, giving a line became a long term bet on the user, and so we looked at long term behavior of the user (features of approval models are derived from last 180 day activity of the user).

However, the unstated assumption is not valid - particularly in the delinquency / retention conscious scenario that we are in.

Thus, immediately we can do the item (1) listed in the summary above.

Note that an alternative do evaluation on D + 30 would be daily evaluation, but in the discussion we agreed that we don't want users to see simpl button one day, and not see it other day. So we will do the evaluation every 30 days.

Analyzing last 30 day activities

Next we come to the question - while assigning limits, how should we use the last 30 day activity?

From March 1 to March 15 we got around 38K NU (on PL). The break up across lines was as following (some users may have multiple lines)

RF_V4:	6K
BATCH_V5:	8.5K
BATCH_V6 (not based on approval days):	2.8K
APPROVAL_DAYS:	15K
Merchant parm based lines (Myntra, Zepto, Zomato):	~10K

If we analyze the performance of num seen days in last 30 days performance across lines, we see a markedly high dlq/low retention only in BATCH_V6 and APPROVAL_DAYS line.

Our rule will be -

At the time of giving the user a line - we will check if they have been seen for at least 3 days in last 30 days.

Implications

If this rule were live then in Mar 1-15 period, instead of acquiring 38K users, we would have acquired around 90% of 38K users. Delinquency would go down by ~60bps (DPD 60) or ~50 bps terminal.

M1 retention would go up by 1.3% (41.9% to 43.2%)

Overall all:

NU down by 10%

DIq down by 50 bps

Retention up by 1.3%

APPENDIX

Here is the aggregate data

past seen days	user dlq	amt dlq	retention
0	14.65	18.31	28.70
1	13.74	17.38	30.69
2	11.78	15.03	34.98
3	11.92	15.60	36.23
4	10.70	14.01	38.80
5	10.41	13.27	39.61
6	11.57	14.48	42.70
7	11.07	14.91	43.64
8	9.98	13.28	46.32
9	10.41	13.63	45.29
10	10.22	12.53	47.87
11	10.02	12.21	47.55
12	9.31	11.97	48.39
13	10.14	13.92	50.08
14	8.64	10.80	50.50
15	11.70	13.90	48.80
16	11.98	13.79	47.68

17	10.42	12.14	48.83
18	10.30	12.33	51.33
19	10.67	13.76	45.78
20	11.54	12.73	50.00
21	10.07	9.36	50.69
22	6.04	6.49	50.57
23	4.35	5.58	48.91
24	10.64	12.55	45.39
25	7.02	6.71	47.37
26	5.77	5.26	54.81
27	7.14	7.56	42.86
28	10.42	12.16	31.25
29	12.50	10.24	41.67
30	6.25	6.04	31.25
Grand Total	11.09	13.94	41.88

Now, out of 38K NU in the first 15 days of March, here is breakup of various lines:

RF_V4 accounted for 6K of these 38K NU, and stats for it are

past seen days	user dlq	amt dlq	retention
0	5.85	5.83	20.33
1	7.35	8.80	24.17
2	3.07	3.41	30.26
3	4.41	5.79	29.41
4	5.40	6.53	30.24
5	3.37	3.85	29.21
6	4.51	6.62	34.20
7	3.70	4.63	31.48
8	4.25	6.62	38.53
9	3.37	2.30	38.04
10	3.01	1.99	37.22
11	3.07	2.35	41.23
12	1.44	3.83	39.71
13	3.43	6.68	41.14
14	2.26	2.55	42.94
15	1.28	1.80	42.95

16	4.76	4.41	37.14
17	2.25	4.38	42.70
18	2.13	2.59	44.68
19	0.00	0.00	39.06
20	3.70	3.03	42.59
21	1.64	0.16	59.02
22	0.00	0.00	45.00
23	0.00	0.00	40.54
24	5.26	13.43	36.84
25	0.00	0.00	40.00
26	0.00	0.00	57.14
27	0.00	0.00	53.33
28	0.00	0.00	0.00
29	0.00	0.00	33.33
30	0.00	0.00	50.00
Grand Total	3.85	4.55	33.83

DLQ for this line is so low that we should not touch it at all.

Here is the stats for Batch_v5. Accounts for 8.5K out of 38K users:

past seen days	user dlq	amt dlq	retention
0	12.40	13.72	21.90
1	9.75	11.89	26.17
2	7.89	9.64	29.28
3	7.09	8.64	28.65
4	5.81	7.82	30.17
5	6.66	9.82	34.80
6	4.36	4.39	39.45
7	6.10	9.06	40.67
8	5.65	8.89	41.96
9	5.01	6.07	39.86
10	5.98	7.66	42.45
11	4.85	5.68	40.45
12	4.81	6.63	44.44
13	5.75	8.23	39.38

14	2.09	1.88	43.46
15	3.05	4.54	52.44
16	7.55	5.54	47.17
17	7.20	7.12	43.20
18	6.06	8.97	43.43
19	5.41	10.06	39.19
20	6.56	6.58	54.10
21	12.24	9.13	48.98
22	0.00	0.00	31.15
23	0.00	0.00	56.25
24	12.00	17.50	56.00
25	0.00	0.00	35.29
26	4.55	3.91	54.55
27	12.50	9.05	43.75
28	0.00	0.00	38.46
29	0.00	0.00	50.00
30	0.00	0.00	50.00
Grand Total	6.71	8.28	35.41

Here we can remove past seen days 0 users.

APPROVAL_DAYS line accounted for 15K / 38K users,

past seen days	user dlq	amt dlq	retention
0	25.62	31.52	34.24
1	28.81	33.31	34.41
2	23.37	29.27	37.82
3	20.66	25.28	44.96
4	19.73	24.15	46.37
5	18.38	21.48	46.11
6	19.34	23.18	49.08
7	17.89	22.79	49.38
8	14.92	18.71	51.68
9	15.30	19.20	49.90
10	15.07	17.67	52.63
11	14.81	17.83	52.06

12	13.49	16.39	53.27
13	14.63	19.02	55.66
14	13.11	16.03	54.90
15	18.42	21.72	50.61
16	16.83	19.39	50.24
17	13.84	16.65	52.54
18	14.50	16.84	55.59
19	14.88	18.39	48.76
20	15.25	17.17	52.47
21	15.11	15.04	48.92
22	12.03	12.75	62.41
23	7.78	9.80	55.56
24	12.35	11.34	44.44
25	13.79	12.75	51.72
26	9.80	8.23	58.82
27	8.70	11.20	39.13
28	26.32	28.85	36.84
29	30.00	27.42	40.00
30	10.00	10.21	30.00
Grand Total	17.42	20.82	48.85

Here we can remove past seen days 0, 1, 2 users.

Batch v6 accounts for 2.8K users. For batch v6, the data is:

past seen days	user dlq	amt dlq	retention
0	20.37	25.65	37.04
1	5.41	7.14	48.65
2	19.57	25.10	41.30
3	15.12	18.03	41.28
4	10.83	9.71	45.86
5	10.27	12.74	45.41
6	10.34	10.87	42.36
7	9.21	9.55	43.10
8	7.18	9.96	48.21
9	9.23	13.97	45.13

10	5.75	6.47	44.25
11	5.23	7.83	47.67
12	8.89	10.29	43.70
13	7.75	10.93	54.26
14	5.56	8.38	52.78
15	10.53	8.80	40.00
16	8.11	10.31	50.00
17	11.54	7.78	44.23
18	9.84	10.36	50.82
19	14.58	20.88	41.67
20	9.76	12.88	36.59
21	3.57	2.02	50.00
22	0.00	0.00	37.04
23	5.00	8.76	30.00
24	15.38	19.75	53.85
25	0.00	0.00	54.55
26	0.00	0.00	33.33
27	0.00	0.00	0.00
28	0.00	0.00	60.00
29	0.00	0.00	50.00
30	0.00	0.00	0.00
Grand Total	9.37	11.19	45.12

Here again we can remove 0, 1, 2.

Similarly, Myntra line contributes 3K users, we can remove 0, 1

FWIW, control set information is:

num seen days	retention	user wise dlq	amt dlq
0	21.21%	42.42%	57.06%
1	33.33%	20.00%	10.20%
2	31.82%	20.45%	24.48%
3	30.23%	25.58%	33.55%
4	42.22%	6.67%	8.09%

5	41.67%	10.42%	11.04%
6	38.10%	14.29%	10.18%
7	40.43%	12.77%	12.63%
8	47.37%	7.89%	14.77%
9	42.11%	7.89%	8.44%
10	43.14%	11.76%	15.23%
11	53.13%	12.50%	14.19%
12	48.48%	15.15%	9.72%
13	52.63%	5.26%	6.03%
14	50.00%	10.71%	14.85%
15	44.00%	16.00%	16.13%
16	45.00%	10.00%	8.98%
17	43.48%	17.39%	29.87%
18	57.14%	4.76%	4.30%
19	43.48%	4.35%	6.33%
20	73.33%	6.67%	4.06%
21	37.50%	0.00%	0.00%
22	83.33%	0.00%	0.00%
23	50.00%	0.00%	0.00%
24	25.00%	25.00%	18.47%
25	80.00%	0.00%	0.00%
26	66.67%	0.00%	0.00%
28	50.00%	0.00%	0.00%
29	100.00%	0.00%	0.00%
30	0.00%	0.00%	0.00%
Grand Total	43.09%	13.30%	14.43%

Source data:

[Dlq and retention by 30d activity](#)

Source query:

```
WITH first_txn_data AS (
-- First get the users who did first txn between March 1 and March 15
  SELECT
    user_id,
    TO_DATE(first_txn_timestamp) AS first_txn_date,
```

```

        TRANSFORM(
            FROM_JSON(first_txn_credit_approvers_gcls, 'array<map<string,string>>'),
            x -> x['approver']
        ) AS gcls
FROM
    public.user_profile
WHERE
    first_txn_product_id = 'a21b31b4-74b8-4a6e-a2a8-28b85d1f8315'
    AND first_txn_timestamp BETWEEN '2025-03-01' AND '2025-03-15'
), user_list AS (
-- Extract user list once to avoid repeated subqueries
    SELECT user_id, first_txn_date FROM first_txn_data
), past_seen_data AS (
-- Find all (user id, date) pair for approval calls between Feb 1 and March 14
-- since we will check approval call in 30 days prior to 1st txn
    SELECT
        uae.user_id,
        COUNT(DISTINCT uae.bucket_id) AS count
    FROM
        public.user_approval_events uae
    INNER JOIN user_list ul ON uae.user_id = ul.user_id
    WHERE
        uae.bucket_id >= 'date_2025_02_01'
        AND uae.bucket_id <= 'date_2025_03_14'
        AND uae.event_name IN (
            'ZeroClickTransactionEligibilityFailedEvent',
            'UserApprovedEvent',
            'UserApprovalFailedEvent',
            'ZeroClickTransactionEligibilitySuccessEvent'
        )
        AND uae.product_id = 'a21b31b4-74b8-4a6e-a2a8-28b85d1f8315'
        AND TO_DATE(SUBSTR(REPLACE(uae.bucket_id, '_', '-'), 6, 10)) BETWEEN
            DATE_SUB(ul.first_txn_date, 30) AND DATE_SUB(ul.first_txn_date, 1)
    GROUP BY uae.user_id
), retention_data AS (
-- For retention we will find any transactions between 11 and 40 days of first
transaction
    SELECT
        te.user_id,
        COUNT(DISTINCT te.bucket_id) AS count
    FROM
        public.transaction_events te
    INNER JOIN user_list ul ON te.user_id = ul.user_id
    WHERE
        te.bucket_id BETWEEN 'date_2025_03_11' AND 'date_2025_04_25'
        AND te.event_name = 'TransactionClaimedEvent'
        AND te.product_id = 'a21b31b4-74b8-4a6e-a2a8-28b85d1f8315'
        AND TO_DATE(SUBSTR(REPLACE(te.bucket_id, '_', '-'), 6, 10)) BETWEEN
            DATE_ADD(ul.first_txn_date, 11) AND DATE_ADD(ul.first_txn_date, 40)
    GROUP BY te.user_id
), settlement_data AS (
    SELECT
        sss.user_id,
        CASE
            WHEN COUNT(sss.settled_status) = 0 THEN 1 -- No settlements found

```

```

        WHEN COUNT(sss.settled_status) = COUNT(CASE WHEN sss.settled_status =
'SETTLED' THEN 1 END) THEN 1
        ELSE 0
    END AS settled_status,
    ROUND(SUM(IF(sss.settled_status = 'SETTLED',
        sss.amount_paid, COALESCE(sss.total_amount_due, 0) -
COALESCE(sss.fine_amount_not_waived_in_paise, 0))) / 100, 0) AS tpv,
    ROUND(SUM(IF(sss.settled_status != 'SETTLED', COALESCE(sss.total_amount_due,
0) - COALESCE(sss.fine_amount_not_waived_in_paise, 0), 0)) / 100, 0) AS dlq_amount

FROM
    public.settlement_summary_from_snapshot sss
    INNER JOIN user_list ul ON sss.user_id = ul.user_id
WHERE
    TO_DATE(sss.cycle_end_date) BETWEEN ul.first_txn_date AND
DATE_ADD(ul.first_txn_date, 15)
    GROUP BY sss.user_id
), credit_limit_data AS (
-- Get the most recent credit limit event before first transaction for each user
    SELECT
        ucl.user_id,
        ucl.event_approver AS last_credit_limit,
        TO_DATE(ucl.event_timestamp) AS credit_limit_date
    FROM
        public.user_credit_limits ucl
    INNER JOIN user_list ul ON ucl.user_id = ul.user_id
    WHERE
        TO_DATE(ucl.event_timestamp) < ul.first_txn_date
        AND ucl.event_name IN ('UserCreatedEvent', 'UserCreditLimitCreatedEvent')
    QUALIFY ROW_NUMBER() OVER (
        PARTITION BY ucl.user_id
        ORDER BY ucl.event_timestamp DESC
    ) = 1
)
)
SELECT
    ftd.user_id,
    ftd.gcls,
    cld.last_credit_limit,
    DATEDIFF(ftd.first_txn_date, cld.credit_limit_date) AS days_since_cl,
    COALESCE(psd.count, 0) AS past_seen_count,
    IF(COALESCE(rd.count, 0) > 0, 1, 0) AS retention_bool,
    COALESCE(sd.settled_status, 1) AS settled_status,
    COALESCE(sd.tpv, 0) AS tpv,
    COALESCE(sd.dlq_amount, 0) AS dlq_amount
FROM
    first_txn_data ftd
    LEFT JOIN past_seen_data psd ON ftd.user_id = psd.user_id
    LEFT JOIN retention_data rd ON ftd.user_id = rd.user_id
    LEFT JOIN settlement_data sd ON ftd.user_id = sd.user_id
    LEFT JOIN credit_limit_data cld ON ftd.user_id = cld.user_id
ORDER BY ftd.user_id

```