

DM_consistency_180d - 설계노트 (Version v1 scope)

1, 목적(Why)

유저의 “방문/구매 리듬(regularity)”을 가입일 기준 180일 윈도우에서 수치화한다.
핵심은 단순 활동량(volume)과 분리된 규칙성(consistency)을 만들고, 이후 LTV/Retention에서 초기 전환(activation)과 장기 성과 사이의 trade-off를 설명하는 주요 설명변수로 사용한다.

2, Grain

1 user_id = 1 row

기준일은 signup_date이며, 모든 세션/주문 기반 지표는 signup_date 기준 0~179일(180d) 윈도우에서 계산된다.

3, Input tables & Join key

- users - 기준 테이블(signup_date), key - user_id
- sessions - 180일 방문 리듬/규칙성 지표(활동일, 방문간격, 주단위 활동) 산출, key - user_id (session_id는 volume 계산용)
- orders - 180일 구매 리듬/재구매 지표(주문 간격, 재구매 플래그) 산출, key - user_id (order_id는 volume/간격 계산용)
- 참고(스캔 최적화): 적용 window가 (0~180)일이므로, signup_date 기준 ****bounds(min_signup, max_signup)****를 계산해 sessions/orders를 $\text{min_signup} \leq \text{date} < \text{max_signup} + 180$ 일로 먼저 필터한 뒤, 최종 집계에서 유저별 window 조건을 다시 적용한다.

4, Partition / Clustering

- **PARTITION BY** : signup_date 코호트/기간 분석이 핵심(“가입 월별” slice)
- **CLUSTER BY** : user_id 후속 DM/분석에서 user join이 대부분

5, Window 정의

- 기준일: signup_date
- 전처리 (global filter, 스캔 절감 목적)
 - $\text{min_signup} \leq \text{date} < \text{max_signup} + 180$ 일 범위로 sessions/orders를 먼저 필터

- 유저별 window(최종 집계 기준)
 - 0-180d: $\text{signup_date} \leq \text{date} < \text{signup_date} + 180\text{일}$ (consistency + volume 계산 윈도우)
- 경계 규칙: 시작일 포함, 종료일 미포함

6, Main Features and 계산 로직

A. Volume controls (세션 기반)

- session_cnt_180d: session_id count, active_days_180d: session_date count

B. Session-based consistency (방문 리듬)

1. inter-visit gap(방문 간격) 지표

- 방문일 시퀀스 생성: `sessions_days = distinct (user_id, session_date)`
- gap 계산: `LAG(session_date)`로 이전 방문일 대비 `DATE_DIFF(day)`
- intervisit_mean_180d: gap 평균
- intervisit_median_180d: gap 중앙값 (`APPROX_QUANTILES` 기반)
- intervisit_std_180d: gap 표준편차(`STDDEV_SAMP`)
- intervisit_cv_180d: gap 변동계수 = std / mean (`SAFE_DIVIDE + NULLIF`로 0/NULL 방지)
-> CV가 클수록 방문 간격이 불규칙(= low consistency)

2. weekly regularity (주 단위 규칙성)

- active_weeks_180d: 180일 내 활동한 주 수 (`DATE_TRUNC(session_date, WEEK(MONDAY))` distinct count)
- weekly_active_ratio_180d: $\text{active_weeks} / (\text{윈도우 내 총 주 수})$
 - 총 주 수는 `GENERATE_DATE_ARRAY`로 `signup` 주부터 `signup+179일`까지 주 단위 `array length`로 계산

C. Orders-based consistency (구매 리듬)

1. 주문 volume + 첫 구매일

- orders_180d: distinct order_id count, first_order_date_180d: 최초 order_date

2. repeat interval(재구매 간격) 지표

- 주문일 시퀀스 생성: order_days = distinct (user_id, order_date)
 - gap 계산: LAG(order_date)로 이전 구매일 대비 DATE_DIFF(day)
 - repeat_interval_mean_180d / std / cv: gap의 평균/표준편차/변동계수
 - o $CV = std / mean$ (SAFE_DIVIDE + NULLIF 사용)
- ### 3. repurchase flags (첫 구매 이후 재구매 여부)
- repurchase_30d / 60d / 90d
 - o 로직: 첫 구매일(first_order_date_180d) 이후, 해당 기간 내 추가 주문 존재 여부
 - o first_order_date가 NULL이면 FALSE 처리(IFNULL)

D. Consistency score (최종 스코어)

- consistency_score_v1 = $z(\text{active_days_180d}) - z(\text{intervisit_cv_180d})$
 - o $z(x) = (x - \text{전체 평균}) / \text{전체 표준편차}$ (window function OVER() 사용)
 - o 해석: 활동일이 많고(active_days↑) 방문 간격 변동이 작을수록(CV↓) 점수가 커짐
 - o 주의: active_days_180d 또는 intervisit_cv_180d의 표준편차가 0이면 NULL 방지(NULLIF)

7, Sanity checks

PK 유일성: row 수 = distinct user_id 수

값 범위:

- session_cnt_180d, active_days_180d, orders_180d ≥ 0

- weekly_active_ratio_180d는 0~1 범위 (NULL 가능)

논리/관계 체크:

- active_days_180d \leq session_cnt_180d (하루에 여러 세션 가능)
- active_weeks_180d \leq ceil(180/7)=약 26주 범위 내 (정확한 계산은 weekly array 기준)
- intervisit 지표는 활동일이 2일 이상인 유저에서만 의미 있음 (gap 1개 이상). 그 외 유저는 mean/std/cv가 NULL일 수 있음.
- orders_180d = 0이면 first_order_date_180d는 NULL이고, repurchase_30/60/90d는 FALSE여야 함.

스코어 계산 점검:

- Consistency_score_v1 - Null check
- extreme outlier(예: intervisit_cv_180d 매우 큰 값) 존재 시 스코어가 과도하게 왜곡되지 않는지 샘플 점검

8, DM이 꼭 필요한지

필요 — 프로젝트 핵심 가설의 중심 변수(**Consistency**)를 **180d** 윈도우로 표준화해 만들고, **volume**과 분리된 규칙성 지표를 **downstream LTV/Retention** 설명에 재사용할 수 있다.