북미 도매전력 시장 가격 변동 원인 분석 파이프라인

Jake Tae-moon Jo · 개발 리드 · https://jake-jo-portfolio.vercel.app/ko/work/grid-constraint-analysis/

송전 제약 분석 메모를 자동으로 만드는 파이프라인을 가설 실험 22건으로 다듬고, 채점 코드가 실제 메모를 만드는 코드와 다른 것을 찾아 채택한 가설을 되돌렸습니다.

가설 6건 채택	29→46%	−39점	+22 · +13점
22건 중(기각 6 · 미결 8 · 입력 채택 2)	선행 트리거 적중률, 가설 1건(dev · train 22→32%)	합의 점수, 운영 경로(train)	정의 분리 효과 (train · dev)

결과

42일 동안 가설 실험 22건을 기록해 6건을 채택하고 6건을 기각했으며, 기각한 하나는 병합 뒤에 되돌린 가설 04입니다.

고객의 수정 요청은 모두 24시간 안에 반영했습니다.

2026년 8월 3일 첫 커밋부터 9월 14일 마지막 푸시까지 42일 동안 가설 실험 22건을 기록했고 6건을 채택, 6건을 기각했습니다. 미결은 8건, 입력으로 채택한 것은 2건입니다.

실험 45로 선행 트리거 적중률이 train 22%에서 32%로, dev 29%에서 46%로 올랐습니다.

채점을 운영과 같은 함수로 바꾸자 train의 메모 합의 점수가 −39점으로 드러났고, 정전 등급의 정의를 나누자 train에서 22점, dev에서 13점 올랐습니다(가설 채택이 아니라 정의를 바꾼 효과입니다). 운영 경로에서 정전 critical 재현율은 2%에서 9%입니다.

train에서 이기고 dev에서 진 가설 5건을 기각했습니다. 실험 49는 dev 순이득이 0인 채로 채택했습니다.

이 CSV 메모를 거래 담당자가 실제로 썼는지는 레포에 기록이 없습니다. 결과는 만든 것과 잰 것까지입니다.

기간
2026.08 – 2026.09

역할
사람은 본인 1명 · AI 에이전트 팀 설계·운영

팀
본인 1명 + AI 에이전트 (커밋 저자는 본인뿐)

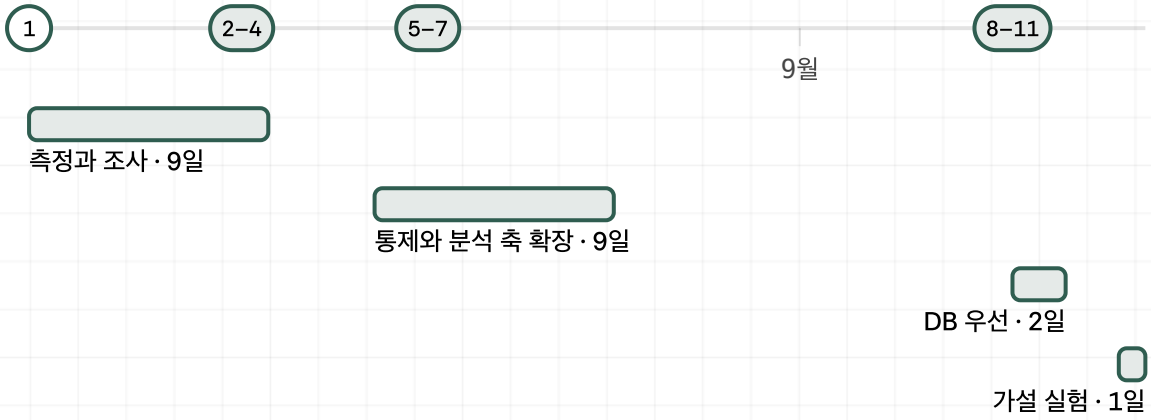
플랫폼
명령줄 분석 파이프라인 · CSV 메모 · 가설 실험 하네스

배경
북미 도매전력 시장에서 송전 제약이 묶인 이유를 거래 담당자가 손으로 분석하던 일을 자동화한 고객사 작업

42일, 가설 실험 22건

42일 동안 가설 실험 22건을 기록해 6건을 채택하고 6건을 기각했으며, 기각 하나는 병합 뒤에 되돌린 가설 04다.

42일	22건	6건
첫 커밋부터 마지막 푸시까지	기록한 가설 실험	채택한 가설



1	2026년 8월 3일	첫 커밋	
2	2026년 8월 11일	거래 담당자 메모 전수 조사	메모 1,482건, 원인 줄 4,853건
3	2026년 8월 12일	이관용 저장소 분리	제약 프로파일 89개를 회귀 기준선으로(8월 13일까지), 서비스 쪽은 미착수
4	2026년 8월 17일	게이트를 못 넘은 CSV 2건 발행	단계를 손으로 돌린 결과
5	2026년 8월 18일	발행 도장	도장과 노트 지문이 맞아야만 발행
6	2026년 8월 19일	단일 진입점·단계표 대조	단계표와 실행 스크립트를 양방향으로 대조, 없던 단계 4개 발견
7	2026년 8월 24일	참조 메모를 읽지 않는 기본값	비교가 순환 논증이 되지 않게
8	2026년 9월 9일	입력은 고객 DB에서 먼저	9월 11일까지 DB 하나로 수집
9	2026년 9월 13일	가설 실험 하네스	사전 등록 → train/dev 채점 → 판정, 보고서 44건(9월 13~14일)
10	2026년 9월 14일	가설 04 되돌림	운영 경로로 재측정해 기각
11	2026년 9월 14일	마지막 푸시	

문제와 배경

거래 담당자가 손으로 쓰던 「왜 묶였나」 메모를, 숫자로 추적되는 설명으로 자동 생성하는 일이었습니다.

북미 도매전력 시장에서는 송전 제약이 실시간 시장에서 묶이면(bind) 가격이 갈립니다. 혼잡 가격 변동을 보는 담당자는 「왜 묶였나」를 손으로 분석해 메모로 남겼습니다. 과제는 그 설명을 숫자로 추적되는 형태로 자동 생성하는 것이었고, 산출물은 제약 한 건당 한 행인 CSV입니다.

제약 한 건마다 세 질문에 숫자로 답해야 했습니다. 어느 가격 노드가 어느 방향으로 얼마나 움직이는지, 원인이 부하·열, 송전 정전, 재생에너지, 인접 회랑, 접경 흐름 중 무엇인지, 하루 전 시장이 이미 알고 있었는지입니다. 하루 전 시장이 이미 가격에 반영한 것은 우위가 아닙니다. 원인은 여섯 가지(마지막은 「지지 원인 없음」)와 하루 전 반영 세 칸의 표로 분류하고, 「지지 원인 없음」은 기본값이 아니라 다른 원인을 시험해 모두 실패한 뒤에 남는 것으로 정했습니다.

참조 정답은 거래 담당자가 손으로 쓴 과거 메모입니다. 8월 11일에 메모 전수 조사를 했고, 메모는 1,482건, 원인 줄은 4,853건이었습니다.

표본이 작았습니다. 제약당 묶임 시간은 중앙값 7시간이고, 192건 중 100시간을 넘는 것은 0건이었습니다. 8월 28일에 회귀·ML·계층 모델은 쓸 수 없고 표본이 필요 없는 결정적 분해만 남는다고 진단했습니다.

제약 한 건이 메모가 되는 길

제약 한 건은 호출 73개와 LLM 분석·반박, 독립 재계산, 게이트 21종을 지나야 CSV 한 행이 되고, 그 행을 시스템에 붙이는 일은 사람이 하도록 정했다.

① **세 질문** □시스템
어느 노드가 얼마나 움직이나 · 원인이 무엇인가 · 하루 전 시장이 이미 알았나

② **하루치 수집** □시스템
고객 DB 우선, 공개 시장·기상 피드는 보조. 필수 입력이 없으면 시작하지 않는다

③ **숫자로 분해** □시스템
정전·부하·재생·인접 회랑·접경 흐름을 호출 73개로, 첫 실패에서 정지

④ **LLM 분석** ◇에이전트
초안 검토, 20개 차선 병렬. 이 단계와 다음 단계가 시간의 약 85%

⑤ **LLM 반박** ◇에이전트
기본 판정은 REFUTED, CONFIRMED는 출처 인용이 있어야

⑥ **독립 재계산** □시스템
과거 오류 8종을 재현하는 시험으로만 통과로 인정

⑦ **게이트와 도장** □시스템
G번호 21종, 통과하면 노트 지문이 든 도장을 쓰고 실패하면 지운다

⑧ 사람 ○사람
설계상 메모란에 자동으로 쓰지 않는다

제 역할

방향과 채택·되돌림은 제가 정했고, 코드와 실험 보고서는 거의 모두 AI 에이전트와 함께 썼습니다.

이 레포의 커밋 저자는 433개 모두 저 한 명이고, 사람 협업자의 커밋·PR·리뷰는 없습니다. 433개 중 422개(97.5%)에 AI 공동 작성 표시가 있고, 실험 보고서 44건의 작성자 칸은 모두 AI를 가리킵니다. 사람과 AI가 얼마나 나눠 썼는지는 기록이 없습니다. 그래서 역할은 「직접 작성」이 아니라 「AI 에이전트 팀 설계·운영」입니다.

사람이 정한 것은 문서에 날짜와 함께 남아 있습니다. 정전 이력은 오래된 캐시를 믿지 않고 새로 받는다(8월 17일), 필수 입력이 모두 있어야 분석을 시작한다(8월 23일), 참조 메모를 읽지 않는 것을 기본으로 한다(8월 24일), critical의 정의와 점수 규칙(9월 2일), 파이프라인은 작은 창부터 고친다(9월 7일), 그리고 9월 14일의 세 가지입니다. 고객 DB에 있는 것은 DB에서 먼저 받는다, 입력을 DB로 바꾼 실험 16을 입력으로 채택한다, 가설 04는 병합 전에 홀드아웃을 한 번 열어 병합했다가 되돌린다.

만든 것은 네 가지입니다. 진입점 하나에서 하루치 수집과 호출 73개, LLM 검토 2단계(분석·반박), 독립 재계산, 게이트 21종, 발행 도장을 거쳐 CSV가 나오는 분석 파이프라인. 소수 원값 표기를 린터가 실패시키는 메모. 메모 구간 여섯 개를 따로 채점하는 가설 실험 하네스. 그리고 파이프라인을 따로 떼어 낸 서비스 이관용 저장소입니다. 분리한 저장소의 파이프라인은 README에 15단계로 적혀 있고, 제약 프로파일 89개를 회귀 기준선으로 씁니다. 서비스 쪽은 시작하지 않았습니다.

이 일은 개발 규칙도 같이 정했습니다. 로그인·인증은 사람만 하고, 시스템의 메모란에는 쓰지 않고 출력만 하며, 파이프라인을 고칠 때는 4~10건 작은 창부터 돌립니다. 이 규칙은 상시 지시로 문서화돼 있습니다.

누가 무엇을 정했나

방향과 채택·되돌림은 본인 한 명이 정했고, 커밋 433개 중 422개에 AI 공동 작성 표시가 있으며, 발행은 게이트와 도장이 맡는다.

본인 1명

8월 23일: 필수 입력이 모두 있어야 분석을 시작한다
8월 24일: 참조 메모를 읽지 않는 것을 기본으로
9월 2일: critical은 하루 전 시장이 몰랐던 놀람, 정확도 52%로 더 많이 판정하는 쪽
9월 7일: 파이프라인을 고칠 때 4~10건 작은 창부터
9월 14일: 홀드아웃을 1회 열람해 가설 04를 병합했다가 되돌림

AI 에이전트 팀

커밋 433개 중 422개(97.5%)에 AI 공동 작성 표시
실험 보고서 44건의 작성자 칸이 모두 AI를 가리킴
제약마다 분석과 반박 2단계, 20개 차선 병렬
사람과 AI의 분담 비율은 기록이 없다

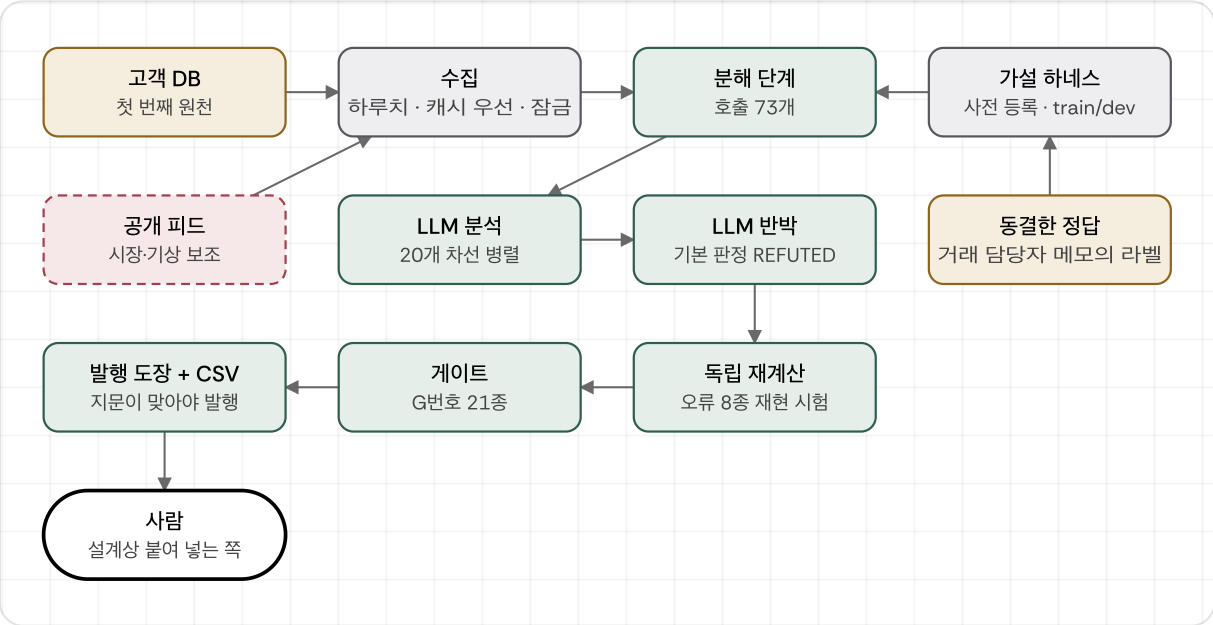
게이트

가설은 사전 등록 → train/dev 비교 → 홀드아웃 1회
게이트 G번호 21종과 발행 도장
단계표와 스크립트의 양방향 대조
PR과 사람 리뷰 기록 없음

시스템 구조

시스템 구성

입력 둘이 하나의 수집 단계로 모여 앞에서 뒤로 한 줄로 흐르고, 가설 하네스는 동결한 정답과 비교해 통과한 변경만 이 줄에 연결한다.



고객 DB

고객 DB에 같은 것이 있으면 DB가 첫 번째 원천이다. 바꾸기 전에 값 하나하나를 교차 확인했다.

수집

하루치를 모으고 캐시를 먼저 쓰며 잠금을 건다. 정전 이력은 오래된 캐시를 믿지 않고 새로 받는다. 필수 입력이 모두 있어야 분석을 시작한다.

분해 단계

사실 확정, 민감도, 정전 매칭·등급(4축 반사실), 부하·기상·램프·예보오차·재생, 흐름·인접 회랑·트리거 노트를 step 61 + 부분 실패 허용 soft 12 호출로 돈다. 첫 실패에서 정지한다.

가설 하네스

가설을 사전 등록하고 변경 3개 파일 이하로 train/dev를 채점한 뒤 채택·기각·미결을 판정한다. 채택한 변경만 정전 등급·조건·선행 트리거 단계에 연결한다. 홀드아웃은 main 병합 때 1회.

공개 피드

DB가 덮지 못하는 날의 대체 입력이다. 공개 시장 피드와 기상·재생·발전 공개 데이터.

LLM 분석

초안 검토 단계다. 20개 차선을 병렬로 돌리고, 이 단계와 다음 단계가 시간의 약 85%를 쓴다.

LLM 반박

반박 모드로 기본 판정이 REFUTED이고 CONFIRMED는 출처 인용이 있어야 한다. 출처 없는 숫자 문장은 주장 장부에 못 들어간다.

동결한 정답

거래 담당자가 손으로 쓴 과거 메모에서 뽑은 라벨을 동결한 것이다. 기본 실행은 이 메모를 읽지 않는다(블라인드).

발행 도장 + CSV

발행 함수는 도장과 지문이 맞지 않으면 파일을 쓰지 않는다. 제약 한 건이 CSV 한 행이다.

게이트

게이트 검사에 G번호 21종이 있다. 통과하면 노트 지문이 든 도장을 쓰고 실패하면 지운다.

독립 재계산

파이프라인과 별개로 다시 계산한다. 과거 오류 8종을 재현하는 시험으로만 통과로 인정한다.

사람

설계상 메모를 사람이 복사해 붙이도록 했다. 시스템에 자동으로 쓰지 않고 로그인도 사람만 한다.

`tools/batch/outage_class.py` · 운영과 채점이 같은 등급 함수를 부르고 달라지는 것은 입력뿐이다. 독스트링에 예전 백테스트가 빠뜨린 검사가 적혀 있다. (독스트링을 줄여 옮김)

```
def grade(t, bind_days, window_days, link_rank, stamps, siblings,
          bind_he, sib_hours=None):
    """Final grade for one outage candidate: (kind, why, confidence).

    Production (`classify`) and scoring (`backtest.score`) both call
    this same function (2026-09-14). The earlier backtest chained its
    own gates in its own order. It skipped the matched control group,
    the planned-outage cap, the rotation probe, the abstain for history
    that was never collected, and the link-rank ceiling.

    What differs between call sites is only the input: the observation
    days, that day's 5-minute stamps, the sibling bind hours."""
    own = _own(bind_days, window_days, t, link_rank, stamps,
               siblings, bind_he, sib_hours=sib_hours)
    kind, why, conf = onset_gate(own["kind"], own["why"],
                                 own["confidence"], ...)
    ...
```

`tools/batch/scorecard.py` · 정답은 +1, 유보는 0, 오답은 −WRONG_COST다. 오답 비용이 새 판정의 정확도 문턱을 정한다. (주석을 줄여 옮김)

```
# The cost of one wrong grade (a correct grade = 1). A new grade only
# earns points once its accuracy clears 1/2.
# The basis for 1 is the 09-02 decision: onset alone was passed over
# for onset-or-necessity (52% precision). Raising it to 2 would push
# the new-grade threshold to 2/3.
WRONG_COST = 1.0

def value(ours, memo, w=None):
    """Row score: correct grade +1, abstain 0, wrong grade -w."""
    if ours not in GRADES:
        return 0.0
    return 1.0 if ours == memo else -(WRONG_COST if w is None else w)
```

핵심 결정

여섯 결정 중 넷은 제가 확정하거나 입력했습니다. 가장 큰 결정인 채점을 운영과 같은 함수로 바꾼 일은 에이전트가 진행했고, 그 결과로 가설 04를 되돌린 것은 제가 정했습니다.

개선의 채택 단위: 제약 id 해시로 60/20/20 분할, 뒤집힌 행의 연결 요소 본인 확정

문제

가설이 정말 나아졌는지 판정하려면 무엇을 한 단위로 세고 데이터를 어떻게 나눌지부터 정해야 했습니다. 표본이 작아 통계가 쉽게 흔들렸습니다.

검토한 선택지

4~10건 파일럿 창을 채택 단위로 (v1): 8건 부분집합을 1,000번 뽑았더니 critical 정밀도 사분위수가 0.33·0.50·0.75로 42점 차가 나서 부적합

코퍼스 연결 요소로 분할 (v6): 거대 요소 하나가 쌍의 33~71%를 삼켜 폐기

✓ 제약 id 해시로 60/20/20 분할, 채택 단위는 뒤집힌 행의 연결 요소, dev는 방향만 확인

결정

제약 id 해시로 train·dev·holdout을 60/20/20으로 나누고, 채택 단위는 같은 제약·티켓·설비로 묶인 뒤집힌 행의 연결 요소로 했습니다. dev는 방향만 확인하고, 검정력 모델로 최소 효과를 정했습니다(train 11행, 이후 14행).

결과

첫 검정력 모델은 행을 독립으로 봐서 검정력이 96%로 부풀었고, 티켓 단위 모델로 고쳤습니다. 가설 22건의 판정에 이 틀을 썼습니다.

대가·남은 위험

dev는 방향 확인뿐이라 일반화 증거가 약합니다. 가설 04의 채택도 dev 요소 4개, p 0.25였습니다. 한 사이클마다 약 79초짜리 백테스트를 두 번 돌립니다.

점수 규칙: 정답 +1 / 유보 0 / 오답 −WRONG_COST 본인 확정

문제

판정을 더 내는 가설이 좋은 가설인지 가르려면, 틀린 판정의 값을 점수에 넣어야 했습니다.

검토한 선택지

정답/비정답 2칸: 유보를 오답으로 바꿔도 잃을 것이 없어 추측이 공짜

✓ 정답 +1, 유보·미채점 0, 오답 −WRONG_COST (WRONG_COST = 1)

결정

행마다 정답 +1, 유보·미채점 0, 오답 −WRONG_COST로 했습니다. 오답 비용이 새 판정의 정확도 문턱(WRONG_COST ÷ (1 + WRONG_COST))을 정하고, 판정을 유보로 되돌려도 같은 이득입니다. 9월 2일에 정확도 52%로 판정을 더 많이 내는 쪽을 택하면서 오답 비용을 1로 두었습니다.

결과

이전 규칙에서는 train의 유보 72건을 supporting으로 바꾸면 45 대 0으로 채택되면서 오답 27건을 발행하게 되는 예가 있었습니다. 규칙을 바꾼 뒤 이 예는 미결(+18 대 최소 효과 27점)로 떨어졌습니다. 가설 03~05는 오답 비용 1과 2에서 판정이 같았습니다.

대가·남은 위험

오답 비용을 2로 올릴지는 열려 있는 결정입니다. 올리면 새 판정에 정확도 67%가 필요합니다.

채점을 파이프라인이 쓰는 같은 함수로 한다 에이전트 기본값 · 확정 기록 없음

문제

가설 04가 채택돼 main에 병합됐는데, 백테스트가 채점하는 경로가 운영 경로와 달랐습니다.

검토한 선택지

백테스트가 후보 순위·흐름 게이트·온셋 게이트를 자기 순서로 이어 채점: 운영이 적용하는 짝 대조군, 사전 공지 상한, 인접 회랑 시험, 이력 미수집 유보, 연결 상한을 빼먹음 (기존)

✓ 백테스트와 운영이 같은 등급 함수를 호출하고 입력만 다르게

결정

정전 등급 함수를 하나로 합쳐 백테스트와 운영이 같은 함수를 부르게 했습니다. 달라지는 것은 입력(관측일, 그날의 5분 단위 시각 등)뿐입니다. 이후 모든 채점은 운영 경로로 합니다.

결과

같은 기준으로 재니 합의 점수가 +4점(51%)에서 -39점(35%, critical 10건 중 5건 정답)으로 내려갔습니다. 채택했던 가설 04는 운영 경로에서 train 1행·dev 1행만 움직였고 둘 다 나빠져 기각하고 되돌렸습니다.

대가·남은 위험

시작점이 -39점이라는 것은 이 모델의 약점입니다.

정전 등급은 물리적 필요성, 사전 공지는 속성으로 따로 적는다 본인 확정

문제

9월 2일의 규칙은 사전 공지된 정전을 critical로 부르지 못하게 했습니다. 한 칸에 「이 정전이 없었으면 묶이지 않았나」와 「하루 전 시장이 알았나」 두 질문이 접혀 있었습니다.

검토한 선택지

9월 2일 규칙: 사전 공지된 정전은 critical이 될 수 없다 (하루 전 시장이 몰랐던 놀람만 critical)

✓ 등급은 물리적 필요성만, 사전 공지 여부는 노트 줄 끝에 별도 사실로

결정

등급은 「이 정전이 없었으면 묶이지 않았나」로만 정하고, 사전 공지 여부는 노트 줄 끝에 따로 적었습니다.

결과

정의를 분리하자 메모 합의 점수가 train에서 22점, dev에서 13점 올랐고, critical 정밀도가 55%에서 64%로 올랐습니다.

대가·남은 위험

정의 결정이라 p 0.068(가설 채택 문턱 0.05 미달)인 채로 적용했습니다. 운영 경로에서 critical 재현율은 2%에서 9%(303건 중 27건)에 그칩니다.

거래 담당자 메모를 읽지 않는 것을 기본으로 (블라인드) 본인 입력

문제

참조 정답이 거래 담당자의 메모인데, 분석이 같은 메모를 읽으면 비교가 순환 논증이 됩니다.

검토한 선택지

메모를 입력으로 쓰고 옵션으로 끈다 (기존 기본값)

✓ 기본 실행은 메모를 전혀 읽지 않고, 옵션으로만 켠다

결정

8월 24일에 기본값을 바꿨습니다. 기존 명령어는 계속 동작하게 남겼습니다.

결과

블라인드가 샌 것이 두 번 실측됐습니다. 환경 변수만 설정해 두면 두 도구가 메모를 읽었고(9월 5일 실측), 입력 덤프 안에도 같은 문장이 들어 있었습니다. 문 세 개(메모 캐시, 입력 덤프의 작성자 블록, 외부 열람)를 금지로 이름 붙이고 인자 플래그로만 넘기게 했습니다.

대가·남은 위험

발전 정전 축은 항상 유보하고 정전 등급은 자체 4축 반사실만으로 서므로, 죽은 축이 있는 노트가 정상 운영이 됩니다. 노트가 짧아진 이유는 사이드카에 남겨야 합니다.

발행은 게이트 도장이 있어야만 한다 에이전트 기본값 · 확정 기록 없음

문제

8월 17일에 게이트를 통과하지 못한 CSV 2건이 발행됐습니다. 단계를 손으로 나열해 돌린 결과였고, 작업 디렉터리 20개 중 14개는 단계 로그가 없었습니다.

검토한 선택지

사람이 각 단계의 종료 코드를 지켜본다 (규칙으로만 존재)

✓ 게이트 통과 시 노트 지문이 든 도장을 쓰고, 발행 함수가 도장과 지문을 확인

결정

게이트가 통과하면 노트 지문이 든 도장을 쓰고 실패하면 지웁니다. 발행 함수는 도장과 지문이 맞지 않으면 파일을 쓰지 않습니다. 시험은 도장 없음, 정상, 승인 뒤 노트 변경, 통과 실패 네 가지입니다.

결과

8월 18일 발행 도장을, 19일 단일 진입점과 단계표 양방향 대조 검사를 넣었습니다. 이 대조가 없던 단계 4개를 찾아냈습니다.

대가·남은 위험

도장은 「통과했다」만 보장합니다. 규칙 문서에 아직 「사람」으로 표시된 항목이 다음 작업 목록입니다.

품질과 검증

train에서 이기고 dev에서 진 가설 5건을 기각했고, 이 한계는 아래에 적었습니다.

품질은 가설 실험 하네스로 쟀습니다. 가설 22건의 판정은 채택 6, 기각 6, 미결 8, 입력으로 채택 2입니다(실험 기록표 22행을 손으로 센 값). 이 페이지의 수치는 모두 실험 보고서와 결과 JSON에 남은 값이고, 이번 작성에서 다시 돌리지 않았습니다.

이 방법이 잡아낸 것은 거짓 개선입니다. train에서 양수였지만 dev에서 음수여서 기각한 가설이 5건이고, train 점수만 봤다면 채택됐을 수 있었습니다. 실험 45(인접 지역 풍력 예보 변수)처럼 두 점수가 함께 오른 경우도 있습니다. 선행 트리거 적중률이 train 22%에서 32%로, dev 29%에서 46%로 올랐고, 순이득은 train +57행, dev

+27행입니다. 바람 조건이 켜진 행의 적중은 103 대 실패 51(67%)이었고 나머지 조건은 36%였습니다.

숨기지 않을 한계가 있습니다. 실험 49(연결 정전이 향후 90일 대부분 예정이면 보류)는 train +59행으로 채택했지만 dev 순이득은 0입니다. 보고서도 효과는 train에서만 나타난다고 적었습니다. 정전 입력을 고객 DB로 바꾼 세 실험은 순이득(행)을 올리지 못했고, 정전 critical 재현율은 운영 경로에서 9%입니다. 합의 점수는 -39점에서 시작했습니다. 미결은 8건입니다.

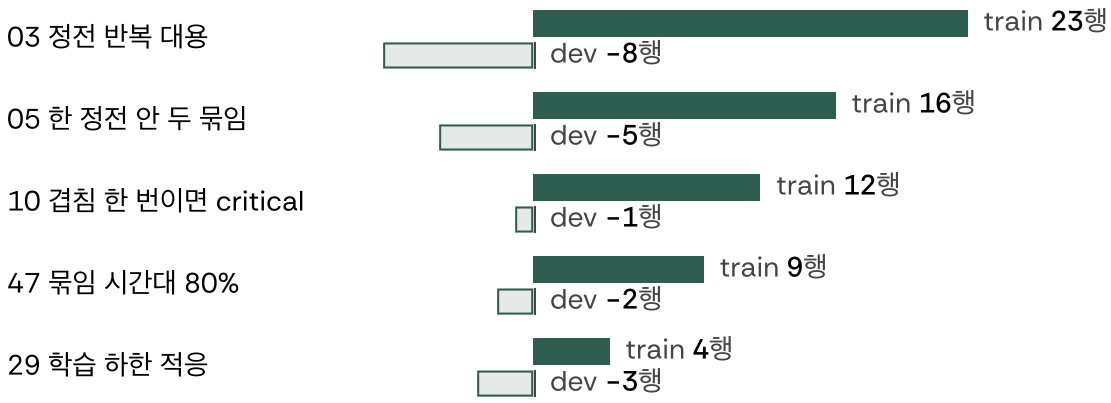
검사는 점검 항목 단위로 쌓았습니다. 입력 계약 19개 검사, 「도구의 침묵은 데이터의 침묵이 아니다」라는 규칙, 과거 오류 8종을 재현해야 통과하는 독립 재계산이 있습니다. 출처 없는 숫자 문장은 주장 장부에 못 들어가고, 장부에 없으면 보고서에 못 갑니다. 단계표와 실행 스크립트는 양방향으로 대조합니다.

테스트는 도구 파이썬 123개 중 110개에 자체 시험 문자열이 있습니다. 이번에 실행 기록을 만들지 않아 통과 수치는 적지 않습니다. 시간은 4~10건 작은 창 한 번이 약 20분이고, 69건 창은 LLM 단계만 1시간을 넘습니다. 같은 창을 다섯 번 돌려 네 번이 한도나 게이트에 끊겼습니다.

train에서 이기고 dev에서 진 가설 5건

train에서는 양수였지만 dev에서 순이득이 음수여서 기각한 가설이 5건이다.

출처: 순이득 = 고친 행 − 깨진 행(실험 레지스트리 사이클 요약표 22행). 이 페이지의 순이득은 모두 이 단위다.



가설 한 건이 지나는 길

가설은 먼저 등록하고 train과 dev로 채점해 판정을 받아야 파이프라인에 연결되며, 같은 가설은 두 번 등록할 수 없다.

① **사전 등록** ○사람
가설과 판정 기준을 JSON으로 먼저 적는다

② **변경** ◇에이전트
파일 3개 이하

③ **train/dev 채점** □시스템
제약 id 해시로 60/20/20 분할, 한 사이클에 약 79초 백테스트 두 번

④ **판정** □시스템
채택 · 기각 · 미결. 채택 단위는 뒤집힌 행의 연결 요소, dev는 방향만 확인

⑤ 파이프라인 연결 □시스템
채택한 변경만 연결

⑥ 홀드아웃 1회 ○사람
main에 병합하기 전에 한 번만 연다

되돌아가는 길
↻ ④ 판정 → ① 사전 등록: 다음 가설 (같은 가설은 두 번 등록할 수 없다)

기능 하나, 처음부터 끝까지

가설 04는 하루 만에 채택되고 병합되고 되돌려졌습니다.

가설 하나가 등록에서 되돌림까지 하루 만에 어떻게 지나갔는지 보여 드립니다. 「승인된 개선이 운영에서 아무것도 바꾸지 않았다는 것을 언제 알았나」가 이 추적의 질문입니다.

가설 04, 채택에서 되돌림까지

옛 경로의 승인은 파이프라인 출력에 영향이 없었다. 쌍 50개 중 3개만 창 판정에 있었고, 채점을 운영과 같게 하자 합의 점수가 +4점에서 −39점으로 내려가 가설 04를 되돌렸다.

+4 → −39점	3쌍(50쌍 중)
합의 점수, 옛 경로에서 운영 경로로	창 판정에 있던 채점 쌍

1–7

채택에서 되돌림까지 · 1일

1	2026년 9월 13일	**가설 04 등록** 겹침이 한 번뿐이고 하루 전 밀도가 낮으면 critical로 부르지 않도록 「20배 밀도」 문턱을 올림
2	2026년 9월 13일	**옛 경로 채점, 채택** train 순이득 +15행(27 대 12), 요소 13 대 2, p 0.002, dev 6 대 2
3	2026년 9월 14일	**홀드아웃 1회 열람, 병합** 고침 9 · 깨짐 6, p 0.34: 같은 방향, p는 문턱 0.05 위
4	2026년 9월 14일	**경로 차이 발견** 쌍 50개 중 3개만 창 판정에 있다
5	2026년 9월 14일	**운영과 같은 함수로 채점** 점수 규칙을 정답 +1 · 유보 0 · 오답 −1로, 합의 점수 +4 → −39점
6	2026년 9월 14일	**운영 경로 재측정, 기각·되돌림** train 1행 · dev 1행만 움직이고 둘 다 나쁨
7	2026년 9월 14일	**정전 등급 정의 분리**

등급은 물리적 필요성, 사전 공지는 속성으로. 합의 점수 train +22 · dev +13점 상승

1 가설 04 등록 (9월 13일)

겹침이 한 번뿐이고 하루 전 밀도가 낮으면 critical로 부르지 않도록 「20배 밀도」 문턱을 올리는 가설입니다.

2 옛 경로 채점, 채택 (9월 13일)

train 순이득 +15행(27 대 12), 요소 13 대 2, p 0.002, dev 6 대 2. 채택했습니다.

3 홀드아웃 1회 열람, 병합 (9월 14일)

main에 병합하기 전에 홀드아웃을 한 번 열었습니다. 고침 9 · 깨짐 6, p 0.34로 같은 방향이지만 유의하지 않았고, 저는 그 상태로 병합했습니다.

4 경로 차이 발견 (9월 14일)

백테스트가 채점한 쌍 50개 중 3개만 운영의 창 판정에 있었습니다. 채점 경로와 운영 경로가 달랐습니다.

5 운영과 같은 함수로 채점 (9월 14일)

점수 규칙을 정답 +1 · 유보 0 · 오답 −1로 바꾸고 운영과 같은 등급 함수로 채점하자 합의 점수가 +4점에서 −39점으로 내려갔습니다.

6 운영 경로 재측정, 기각·되돌림 (9월 14일)

가설 04를 운영 경로에서 다시 재니 train 1행과 dev 1행만 움직였고 둘 다 나빠졌습니다. 기각하고 되돌렸습니다.

7 정전 등급의 정의 분리 (9월 14일)

등급은 물리적 필요성으로, 사전 공지 여부는 속성으로 나눴습니다. 합의 점수가 train에서 22점, dev에서 13점 올랐습니다.

회고

옛 경로의 승인은 파이프라인 출력에 아무 영향이 없었고, 그 사실은 채점기를 운영과 같게 만든 뒤에야 드러났습니다.

승인한 개선이 파이프라인 출력에는 영향이 없었습니다. 가설 04는 홀드아웃이 유의하지 않았는데도(p 0.34) 병합했고, 백테스트가 운영과 다른 경로를 채점하고 있다는 것은 채점기를 운영과 같게 만든 뒤에야 드러났습니다. 이후에는 모든 채점을 운영 경로로 합니다.

8월 17일에는 단계를 손으로 나열해 돌려 게이트를 못 넘은 CSV 2건이 발행됐습니다. 규칙은 문서에 있었지만 검사가 없었습니다. 발행 도장과 단계표 양방향 대조로 바꿨고, 스킬 문서가 1,523줄까지 늘어난 것도 실행 순서·규칙과 검사·사고 기록 세 문서로 나눴습니다.

속이 빈데도 통과한 일이 있었습니다. 묶임 일자 233개가 모두 「지지 원인 없음」인데 검사를 통과했고, 린터가 뜨지 않아 「위반 0」이 나왔으며, 이동 구간 피드가 5일 적은데 파일명이 같아 임계 판정이 뒤집혔습니다. 그래서 입력 계약 검사와 「도구의 침묵은 데이터의 침묵이 아니다」 규칙을 만들었습니다.

확인 없이 쓴 설명이 하루에 두 번 틀렸습니다. 이후 상태를 모두 적고 답하는 규칙을 두었습니다. 큰 창(69건)을 반복해 돌리다 한도에 끊긴 일은 작은 창부터 고치는 규칙으로 이어졌습니다.

아직 열려 있는 것이 있습니다. 미결 가설 8건, 오답 비용을 2로 올릴지의 결정, 운영 경로에서 9%에 그치는 정전 critical 재현율, dev 순이득이 0인 실험 49가 그렇습니다.

일하는 방식

이 프로젝트도 역할별 AI 에이전트 팀으로 만들었습니다. 팀 구성과 게이트는 프로젝트마다 다르고, 이 프로젝트의 것은 위 「제 역할」에 적었습니다. 아래는 EWA 플랫폼을 만들며 정리한 방식입니다.

2026년 EWA 플랫폼을 만들며 세우고 운영한 방식 보기 →

https://jake-jo-portfolio.vercel.app/ko/process/

연락처

whxoans@gmail.com
+82 10-9233-7440