정령계에서 온 커피 연구원

집 에스프레소 머신 옆에서, 한 잔마다 가설을 세우고 스스로 레시피를 고쳐 가는 AI 바리스타 · 분야: 생활 속 AI 활용

1. 집에서 에스프레소를 내리면 생기는 불편

에스프레소는 분쇄도 한 칸, 도징량 0.4 g, 탬핑 힘 하나로 맛이 바뀝니다. 카페에서는 바리스타가 매일 아침 이 값을 맞춥니다(다이얼링 인). 집에서는 하루 두세 잔만 내리니 그라인더는 늘 식은 채로 시작하고, 어제 무엇을 바꿨는지도 잊어버립니다. 그래서 **기록하고, 원인을 가르고, 다음 시도를 제안하는 일**을 AI에게 맡겼습니다. 사람은 다이얼을 돌리고 맛을 판정하고, 나머지는 AI가 합니다.

2. AI는 이렇게 일합니다

AI에게는 "정령계에서 파견된 커피 연구원"이라는 페르소나를 주어, 실험 노트를 쓰는 연구원처럼 일하게 했습니다. 한 잔이 5단계로 흘러갑니다.



그림 1. 5단계 흐름. 사람이 멈추는 곳은 다이얼을 맞출 때와 포터필터를 장착할 때 두 곳입니다.

- **계획**: 그 원두의 *과거 기록*을 읽고 "이번엔 이런 맛을 노리겠다"는 가설과 예상 결과를 적습니다.
- **분쇄·확인**: 실제로 간 도징량과 분쇄 유량이 정상인지 다시 봅니다.
- **추출**: 압력 프로파일대로 뽑고 목표 무게에서 끊습니다.
- **평가**: 결과를 원인 A~F로 분류합니다(A 레시피 / B 온도·압력 / C 유량·시간 / D 채널링 / E 실행은 정상이나 취향 불일치 / F 확신 없음).
- **다음 잔**: "분쇄를 한 칸 조여 보자" 같은 한 줄 제안을 남깁니다.

비용도 신경 썼습니다. 코드가 계산한 신호(의도와 실제 불일치, 온도 불안정, 채널링 감지)가 걸린 잔만 신중한 모델이 다시 분석하고, 정상인 잔은 빠른 모델이 처리합니다.

3. AI가 무슨 생각을 했는지 화면에 보입니다

AI의 판단은 모두 저장되지만, 처음에는 화면에 답변 한 줄만 나왔습니다. 그래서 생각의 흐름을 펼쳐 보여 주는 화면을 만들었습니다. 긴 근거는 접어 두고, 필요하면 펼칩니다.



그림 2. AI 판단 화면(성공한 잔). 계획 → 분쇄 확인 → 추출 → 평가 → "다음 잔은 이렇게". 화면 상단 띠는 데모(시뮬레이션) 표시입니다.

아래는 AI가 계획한 압력 프로파일(점선)과 실제 곡선(실선)을 겹친 화면입니다. 계획은 30초까지였는데 펌프는 무게 컷오프로 29.8초에 멈추는 것처럼, 숫자만 봐서는 놓칠 일이 곡선에서는 한눈에 보입니다.

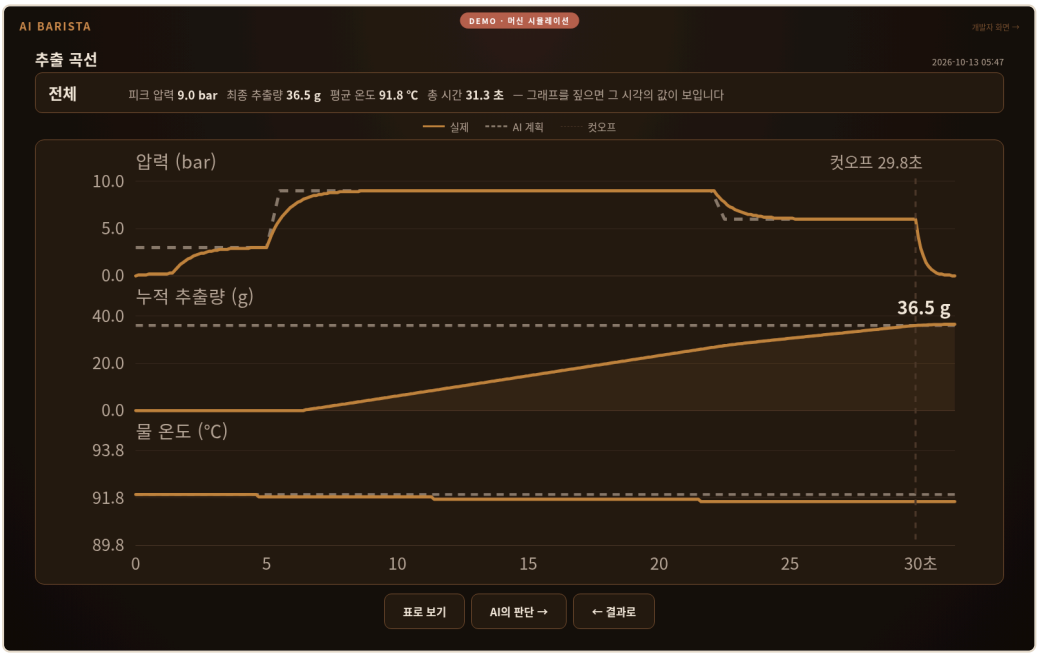


그림 3. 추출 곡선: 압력(계획 대 실제), 누적 무게, 물 온도.

밝혀 둘 점 · 현재 추출 곡선은 **시뮬레이터가 만든 값**입니다. 머신 개조(Gaggiuino)는 부품 도착을 기다리는 중입니다. 분쇄와 도징 측정, 다이얼 기록은 **실물**입니다.

4. 실물 부분: 2만 원 안쪽으로 만든 무게 기반 분쇄

고급 카페 그라인더에는 무게를 재며 갈다가 목표에서 멈추는 기능(Grind-by-Weight)이 있습니다. 같은 구조를 로드셀, 증폭기(HX711), ESP32, 스마트 플러그로 직접 만들었습니다(부품값은 합쳐서 2만 원 안쪽, 저울 노이즈 ±0.03 g). 220V 배선을 직접 만지지 않으려고 전원 제어는 스마트 플러그로 했습니다.

컵 무게를 확인하고, 영점을 잡고, 목표의 대부분을 한 번에 간 뒤 남은 양을 짧은 펄스로 채웁니다. 목표 18.0 g으로 고정한 뒤 최근 4잔의 실측은 17.99~18.34 g이었습니다(오차 +0.34 g 이내). 목표 범위 제한, 중복 실행 거부, 호퍼가 비었을 때

정지, 연락이 끊기면 플러그가 스스로 꺼지는 안전 장치는 실제로 겪은 사고에서 나왔습니다.

5. 실제로 다이얼을 잡아 간 기록

브라질 세하도 원두 한 배치로 19잔을 직접 기록했습니다 (WDT·정압 탬퍼 사용).

다이얼	샷	시간
22	7	21~26초
21	11	19.5~34초
20	1	48초

다이얼 21, 도징 18.0 g, 첫 방울 약 3초, 37~38 g에서 **맛이 좋은 기준 레시피**를 잡았습니다. 20은 한 칸에 20초 넘게 늘어난 넘어간 칸이었습니다. 같은 21에서도 도징량 (18.0~18.7 g)과 원두 숙성에 따라 시간이 흔들렸고, 그 흔들림의 원인을 가르는 것이 AI 기록의 일입니다.

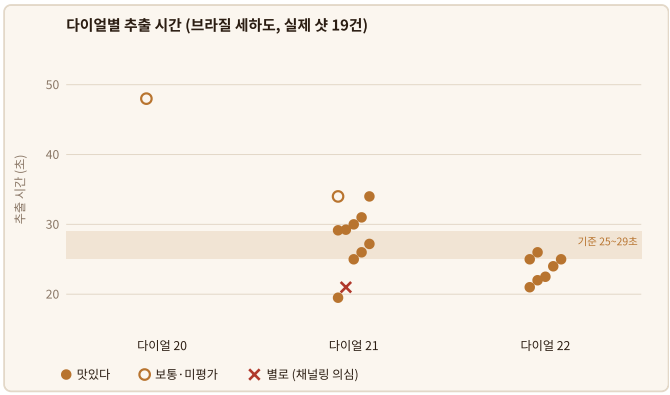


그림 4. 다이얼별 추출 시간(실제 샷).

6. 실패에서 배운 것

- **분쇄 판정이 자기 자신과 비교하고 있었습니다.** 기대 시간과 실제 시간이 같은 공식에서 나와 항상 "정상"이었습니다. 실측을 붙이자 기준 상수가 47% 틀렸다는 것이 드러났고, 이제는 같은 원두·같은 눈금의 과거 *실측 유량*과 비교하며 기준이 없으면 판정을 보류합니다. 없는 기준을 만들어 주면 AI는 그것을 믿습니다.
- **조용히 가짜 모드로 떨어진 사고.** API 키 파일 형식이 바뀐 날 예외 없이 시험용(Mock) 모드로 돌아가 정상처럼 보였습니다. 이제 시작 로그에 그 사실이 보입니다.
- **첫 분쇄 효과.** 식은 분쇄기의 첫 분쇄는 유량이 20~40% 낮았습니다. AI가 "첫 분쇄인가"를 알아야 하는 이유가 측정으로 확인됐습니다.

7. 기계는 재현하고, 사람은 판정한다

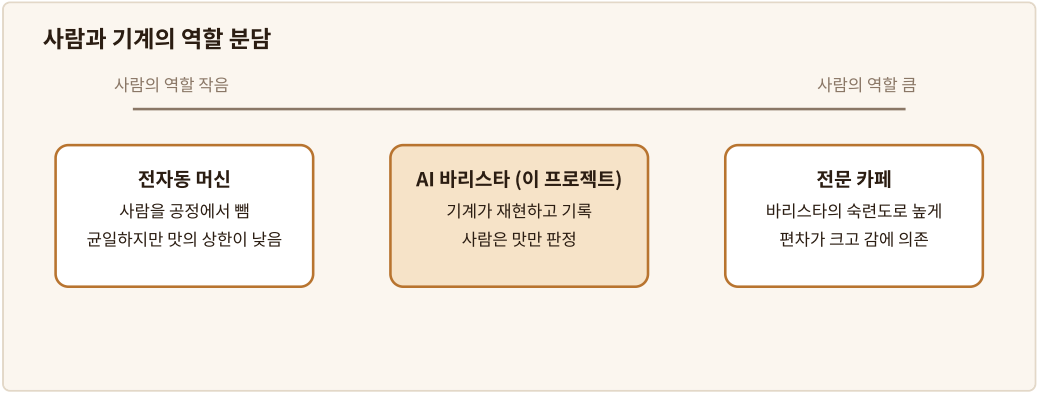


그림 5. 사람과 기계의 역할 분담.

전자동 머신은 사람을 빼서 균일하지만 맛의 상한이 낮고, 카페는 숙련도로 높지만 편차가 큽니다. 이 프로젝트는 그 사이에서, 기계가 재현과 기록을 맡고 **사람은 맛만 판정**하게 합니다. AI가 배울 목표는 표준 수치가 아니라 사람의 판정입니다.

8. 직접 써 보니 이렇게 달라졌습니다

- **기록이 남습니다.** 예전에는 "어제 어땠더라"를 기억에 의존했습니다. 이제 잔마다 다이얼, 도징량, 첫 방울, 시간, 양, 맛이 쌓입니다. 한 배치 원두로 직접 기록한 샷이 19잔입니다.

- **도징이 일정해졌습니다.** 목표를 18.0 g으로 고정한 뒤 4잔의 실측이 17.99~18.34 g이었습니다.
- **시간이 흔들리는 이유를 숫자로 말할 수 있습니다.** 도징 0.4 g 차이가 약 5초를 만들고, 첫 분쇄는 유량이 20~40% 낮고, 다이얼 20은 한 칸에 20초 넘게 늘어난다는 것을 기록에서 찾았습니다.
- **맛을 잡아 가는 길이 짧아졌습니다.** 다이얼 22에서 21로 한 칸 내린 것이 더 좋은 맛으로 이어졌고, 20은 "넘어간 칸"이라는 것을 한 번의 시도로 확인했습니다.

위 숫자는 모두 실제로 내린 샷의 기록입니다. 추출 곡선과 AI의 평가 화면은 시뮬레이터 값이라는 점은 앞에서 밝힌 대로입니다.

앞으로: Gaggia Classic Pro를 오픈소스 Gaggiuino로 개조해 추출 곡선을 실물 값으로 바꾸고, 다이얼 자동 조절과 우유 스팀 온도 기록으로 넓힐 계획입니다.