

AMAZON COMMAND SYSTEM

Amazon 데이터셋 기반 추천 시스템 설계 프로젝트

프로젝트 개요

해당 프로젝트는 Amazon 데이터셋을 기반으로 고객 맞춤형 추천 시스템을 설계하고 구현하는 작업입니다.

Amazon은 전세계적으로 가장 큰 전자상거래 플랫폼 중 하나로, 방대한 고객 리뷰와 제품 데이터를 보유하고 있습니다. 이번 프로젝트에서는 이러한 데이터를 SQL만을 사용하여 분석하고, 고객의 구매 경험을 향상시킬 수 있는 추천 시스템의 기초를 설계하는 것이 목표입니다.

데이터셋 소개

데이터셋 링크 : [Amazon Sales Dataset](https://www.kaggle.com/datasets/karkavelrajaj/amazon-sales-dataset)

Amazon Sales Dataset

This dataset is having the data of 1K+ Amazon Product's Ratings and Reviews



<https://www.kaggle.com/datasets/karkavelrajaj/amazon-sales-dataset>

데이터셋 구조

필드명	설명
product_id	제품의 고유 식별자
product_name	제품의 이름
category	제품이 속한 카테고리 정보
discounted_price	할인된 가격
actual_price	정가
discount_percentage	할인율
rating	제품의 평균 평점
rating_count	평점에 투표한 사람 수
about_product	제품에 대한 간단한 설명
user_id	리뷰를 작성한 사용자의 고유 식별자 (선택으로 구분된 여러 사용자)
user_name	리뷰를 작성한 사용자 이름 (선택으로 구분된 여러 사용자)
review_id	리뷰의 고유 식별자 (선택으로 구분된 여러 리뷰)
review_title	리뷰 제목 (선택으로 구분된 여러 리뷰 제목)
review_content	리뷰 내용 (선택으로 구분된 여러 리뷰 내용)
img_link	제품 이미지 URL
product_link	제품 페이지 URL

데이터 샘플 (LIMIT 10)

[bquxjob_72c707d5_19912527148.csv](#)

데이터 샘플 점검 결과

- category
 - 전부 `|`가 들어있고, 양옆 공백 없이 `"A|B|C"` 형태
- 리뷰 5개 컬럼 (user_id, user_name, review_id, review_title, review_content)
 - 구분자는 행마다 `,`와 `"`가 섞여 있음
 - 선택 주변 공백 통일 후 `SPLIT(',')` 으로 전개
 - `SAFE_OFFSET` 을 활용하여 최대 길이를 기준으로 한 인덱싱 진행하여 누락/오버런 없이 전개 예정

- 할인
 - `discount_percentage` 는 비율 형태로 아래와 같이 2가지로 나누어 제공
 - `discount_frac(0~1)` : **0.48**
 - `discount_pct(0~100)` : **48**
- 링크 컬럼(`img_link`, `product_link`)는 분석 팩트에서 활용이 불가능하여 제외

생성된 테이블

▼ 정제 스테이징

```
CREATE OR REPLACE TABLE `long-ceiling-470102-p4.amazon.products_stg`
CLUSTER BY product_id AS
SELECT
  product_id,
  TRIM(product_name) AS product_name,
  TRIM(category) AS category,
  TRIM(about_product) AS about_product,

  -- 이미 FLOAT 형이라고 가정: 그대로 통과
  discounted_price,
  actual_price,
  discount_percentage AS discount_frac,
  ROUND(discount_percentage * 100, 1) AS discount_pct,
  rating,

  -- rating_count만 FLOAT64로 정규화 (문자 'nan'도 NULL 처리)
  CASE
    WHEN LOWER(CAST(rating_count AS STRING)) = 'nan' THEN NULL
    ELSE CAST(rating_count AS FLOAT64)
  END AS rating_count,

  -- 리뷰 원본(전개용)
  user_id,
  user_name,
  review_id,
  review_title,
  review_content,

  -- 링크는 보존(분석 팩트에서 제외)
  img_link,
  product_link
FROM `long-ceiling-470102-p4.amazon.data`;
```

products_stg 쿼리 다음에서 열기

스키마 세부정보 미리보기 테이블 탐색기 **프리뷰** 통계 계보

필터 속성 이름 또는 값 입력

<input type="checkbox"/> 필드 이름	유형	모드	키	대조	기본값
<input type="checkbox"/> product_id	STRING	NULLABLE	-	-	-
<input type="checkbox"/> product_name	STRING	NULLABLE	-	-	-
<input type="checkbox"/> category	STRING	NULLABLE	-	-	-
<input type="checkbox"/> about_product	STRING	NULLABLE	-	-	-
<input type="checkbox"/> discounted_price	FLOAT	NULLABLE	-	-	-
<input type="checkbox"/> actual_price	FLOAT	NULLABLE	-	-	-
<input type="checkbox"/> discount_frac	FLOAT	NULLABLE	-	-	-
<input type="checkbox"/> discount_pct	FLOAT	NULLABLE	-	-	-
<input type="checkbox"/> rating	STRING	NULLABLE	-	-	-
<input type="checkbox"/> rating_count	FLOAT	NULLABLE	-	-	-
<input type="checkbox"/> user_id	STRING	NULLABLE	-	-	-
<input type="checkbox"/> user_name	STRING	NULLABLE	-	-	-
<input type="checkbox"/> review_id	STRING	NULLABLE	-	-	-
<input type="checkbox"/> review_title	STRING	NULLABLE	-	-	-
<input type="checkbox"/> review_content	STRING	NULLABLE	-	-	-
<input type="checkbox"/> img_link	STRING	NULLABLE	-	-	-
<input type="checkbox"/> product_link	STRING	NULLABLE	-	-	-

▼ 카테고리 브릿지

```
CREATE OR REPLACE TABLE `long-ceiling-470102-p4.amazon.product_categories`
CLUSTER BY product_id, category AS
SELECT
  product_id,
  cat AS category
FROM `long-ceiling-470102-p4.amazon.products_stg`,
UNNEST(SPLIT(REGEXP_REPLACE(category, r'\s*\|s*', '|'), '|')) AS cat
WHERE NULLIF(TRIM(cat), '') IS NOT NULL;
```

product_cate... 쿼리 다음에서 열기

스키마 세부정보 미리보기 테이블 탐색기 **프리뷰** 통계 계보

필터 속성 이름 또는 값 입력

<input type="checkbox"/> 필드 이름	유형	모드	키	대조	기본값	정책
<input type="checkbox"/> product_id	STRING	NULLABLE	-	-	-	-
<input type="checkbox"/> category	STRING	NULLABLE	-	-	-	-

▼ 리뷰 전개

```
CREATE OR REPLACE TABLE `long-ceiling-470102-p4.amazon.product_reviews`
CLUSTER BY product_id, user_id AS
WITH split AS (
  SELECT
    product_id,
    -- 쉼표 주변 공백 표준화 후 split
    SPLIT(REGEXP_REPLACE(COALESCE(user_id, ''), r'\s*\|s*', '|'), '|') AS user_id_arr,
    SPLIT(REGEXP_REPLACE(COALESCE(user_name, ''), r'\s*\|s*', '|'), '|') AS user_name_arr,
    SPLIT(REGEXP_REPLACE(COALESCE(review_id, ''), r'\s*\|s*', '|'), '|') AS review_id_arr,
    SPLIT(REGEXP_REPLACE(COALESCE(review_title, ''), r'\s*\|s*', '|'), '|') AS review_title_arr,
```

```

        SPLIT(REGEXP_REPLACE(COALESCE(review_content, ''), r'\s*,\s*', ','), ',') AS review_content_arr
    FROM `long-ceiling-470102-p4.amazon.products_stg`
),
lens AS (
    SELECT
        product_id,
        user_id_arr, user_name_arr, review_id_arr, review_title_arr, review_content_arr,
        ARRAY_LENGTH(user_id_arr) AS n_uid,
        ARRAY_LENGTH(user_name_arr) AS n_uname,
        ARRAY_LENGTH(review_id_arr) AS n_rid,
        ARRAY_LENGTH(review_title_arr) AS n_rtitle,
        ARRAY_LENGTH(review_content_arr) AS n_rcontent,
        LEAST(
            ARRAY_LENGTH(user_id_arr),
            ARRAY_LENGTH(user_name_arr),
            ARRAY_LENGTH(review_id_arr),
            ARRAY_LENGTH(review_title_arr)
        ) AS n_core
    FROM split
),
expanded AS (
    SELECT
        l.product_id,
        pos,
        l.user_id_arr[SAFE_OFFSET(pos)] AS user_id,
        l.user_name_arr[SAFE_OFFSET(pos)] AS user_name,
        l.review_id_arr[SAFE_OFFSET(pos)] AS review_id,
        l.review_title_arr[SAFE_OFFSET(pos)] AS review_title,
        l.review_content_arr[SAFE_OFFSET(pos)] AS review_content
    FROM lens AS l,
    UNNEST(GENERATE_ARRAY(0, l.n_core - 1)) AS pos
    WHERE l.n_core > 0
)
SELECT
    product_id,
    NULLIF(TRIM(user_id), '') AS user_id,
    NULLIF(TRIM(user_name), '') AS user_name,
    NULLIF(TRIM(review_id), '') AS review_id,
    NULLIF(TRIM(review_title), '') AS review_title,
    NULLIF(review_content, '') AS review_content
FROM expanded
-- 추천/분석의 안정성을 위해 핵심키 존재 보장
WHERE NULLIF(TRIM(user_id), '') IS NOT NULL
    AND NULLIF(TRIM(review_id), '') IS NOT NULL
QUALIFY ROW_NUMBER() OVER (
    PARTITION BY product_id, review_id
    ORDER BY pos
) = 1;

```

product_revie... 쿼리 다음에서 열기

< **스키마** 세부정보 미리보기 테이블 탐색기 **프리뷰** 통계 계보 >

필터 속성 이름 또는 값 입력

<input type="checkbox"/>	필드 이름	유형	모드	키	대조	기본값	
<input type="checkbox"/>	product_id	STRING	NULLABLE	-	-	-	-
<input type="checkbox"/>	user_id	STRING	NULLABLE	-	-	-	-
<input type="checkbox"/>	user_name	STRING	NULLABLE	-	-	-	-
<input type="checkbox"/>	review_id	STRING	NULLABLE	-	-	-	-
<input type="checkbox"/>	review_title	STRING	NULLABLE	-	-	-	-
<input type="checkbox"/>	review_content	STRING	NULLABLE	-	-	-	-

▼ 고객 수/리뷰 수 집계

```
CREATE OR REPLACE TABLE `long-ceiling-470102-p4.amazon.product_reviewers` AS
SELECT
  product_id,
  COUNT(DISTINCT user_id) AS reviewer_count,
  COUNT(DISTINCT review_id) AS review_count
FROM `long-ceiling-470102-p4.amazon.product_reviews`
GROUP BY product_id;
```

product_revie... 쿼리 다음에서 열기

< **스키마** 세부정보 미리보기 테이블 탐색기 **프리뷰** 통계 계보 >

필터 속성 이름 또는 값 입력

<input type="checkbox"/>	필드 이름	유형	모드	키	대조	기본값	
<input type="checkbox"/>	product_id	STRING	NULLABLE	-	-	-	-
<input type="checkbox"/>	reviewer_count	INTEGER	NULLABLE	-	-	-	-
<input type="checkbox"/>	review_count	INTEGER	NULLABLE	-	-	-	-

▼ 분석용 팩트

```
CREATE OR REPLACE TABLE `long-ceiling-470102-p4.amazon.products_analytics`
CLUSTER BY product_id AS
WITH p AS (
  SELECT
    product_id,
    product_name,
    about_product,
    discounted_price,
    actual_price,
    discount_frac,
    discount_pct,
    -- rating: STRING → FLOAT64
    SAFE_CAST(NULLIF(rating, '')) AS rating_f,
    -- rating_count: 'nan' 방지
    SAFE_CAST(NULLIF(LOWER(CAST(rating_count AS STRING)), 'nan')) AS rating_count_f
  FROM `long-ceiling-470102-p4.amazon.products_stg`
),
params AS (
  SELECT
    AVG(rating_f) AS C,
    GREATEST(1.0, LEAST(5.0,
      CAST(APPROX_QUANTILES(rating_count_f, 100)[OFFSET(50)] AS FLOAT64)
    )) AS m
```

```

FROM p
)
SELECT
  p.product_id,
  p.product_name,
  p.about_product,
  p.discounted_price,
  p.actual_price,
  p.discount_frac,
  p.discount_pct,

  -- 정규화된 노출 컬럼
  p.rating_f      AS rating,
  /* ↓↓↓ 예외 처리: reviewer_count=1 && rating 존재 && rating_count NULL → rating_count를 rating으로 대체 */
  CASE
    WHEN p.rating_f IS NOT NULL
      AND p.rating_count_f IS NULL
      AND r.reviewer_count = 1
    THEN p.rating_f
    ELSE p.rating_count_f
  END            AS rating_count,

  r.reviewer_count,
  r.review_count,

  SAFE_DIVIDE(p.discounted_price, NULLIF(p.actual_price, 0)) AS price_ratio,

  /* 베이지안(소 m): 보정된 rating_count 사용 */
  CASE
    WHEN p.rating_f IS NULL THEN NULL
    ELSE (
      (p.rating_f *
        CASE
          WHEN p.rating_f IS NOT NULL
            AND p.rating_count_f IS NULL
            AND r.reviewer_count = 1
          THEN p.rating_f
          ELSE p.rating_count_f
        END
      + params.C * params.m)
      /
      (
        COALESCE(
          CASE
            WHEN p.rating_f IS NOT NULL
              AND p.rating_count_f IS NULL
              AND r.reviewer_count = 1
            THEN p.rating_f
            ELSE p.rating_count_f
          END, 0
        ) + params.m
      )
    )
  END AS bayesian_rating_small_m

FROM p
CROSS JOIN params
LEFT JOIN `long-ceiling-470102-p4.amazon.product_reviewers` r
USING (product_id);

```

products_anal... 쿼리 다음에서 열기

< 스키마 세부정보 미리보기 테이블 탐색기 **프리뷰** 통계 계보 >

필터 속성 이름 또는 값 입력

<input type="checkbox"/> 필드 이름	유형	모드	키	대조
<input type="checkbox"/> product_id	STRING	NULLABLE	-	-
<input type="checkbox"/> product_name	STRING	NULLABLE	-	-
<input type="checkbox"/> about_product	STRING	NULLABLE	-	-
<input type="checkbox"/> discounted_price	FLOAT	NULLABLE	-	-
<input type="checkbox"/> actual_price	FLOAT	NULLABLE	-	-
<input type="checkbox"/> discount_frac	FLOAT	NULLABLE	-	-
<input type="checkbox"/> discount_pct	FLOAT	NULLABLE	-	-
<input type="checkbox"/> rating	FLOAT	NULLABLE	-	-
<input type="checkbox"/> rating_count	FLOAT	NULLABLE	-	-
<input type="checkbox"/> reviewer_count	INTEGER	NULLABLE	-	-
<input type="checkbox"/> review_count	INTEGER	NULLABLE	-	-
<input type="checkbox"/> price_ratio	FLOAT	NULLABLE	-	-
<input type="checkbox"/> bayesian_rating_small_m	FLOAT	NULLABLE	-	-

▼ 카탈로그

```
CREATE OR REPLACE TABLE `long-ceiling-470102-p4.amazon.catalog_light` AS
SELECT product_id, product_name, category, about_product
FROM `long-ceiling-470102-p4.amazon.products_stg`;
```

product_cate... 쿼리 다음에서 열기

< 스키마 세부정보 미리보기 테이블 탐색기 **프리뷰** 통계 계보 >

필터 속성 이름 또는 값 입력

<input type="checkbox"/> 필드 이름	유형	모드	키	대조	기본값	정책 E
<input type="checkbox"/> product_id	STRING	NULLABLE	-	-	-	-
<input type="checkbox"/> category	STRING	NULLABLE	-	-	-	-

1. 추천 시스템 이름

→ "리뷰를 많이 남긴 상품들이에요" / "최대 할인율 기반 추천 시스템"

2. 추천 시스템의 테마: 추천 시스템의 고유 컨셉에 대한 설명

→ 고객이 최대한의 할인율을 적용받을 수 있도록 할인율과 평점을 결합한 추천을 제공합니다.

3. 구현 로직: SQL 쿼리 설명 및 주요 로직 설명

→ SQL 코드 첨부

4. 결과

: 추천 결과의 샘플을 출력하고, 이에 대해서 간단하게 설명해주세요.

→ 빅쿼리 화면을 캡처하여 이미지를 제출하고, 이에 대해서 간단히 설명해주세요.

추천 시스템

1. Modu Box (오늘의 특가 상품)

테마

최소한의 품질을 보장하면서 최대 할인을 제공하는 일일 특가 상품 (쿠팡 골드 박스 모티브)

구현 로직

S (상황)

- 정가 대비 실구매가가 낮은 상품 & 평점/리뷰를 통한 최소 품질을 보장하도록 필터링

T (과제)

- 큰 할인율, 가격 타당성, 기본 품질을 동시에 만족하는 상위 20개 상품 추출

A (행동)

- `products_analytics`, `product_reviewers` 를 `product_id` 로 JOIN
 - 리뷰어에 대한 데이터를 활용해야하기 때문에
- `p.price_ratio BETWEEN 0 AND 0.80`
 - 정가의 80% 이하인 가격으로 할인 최소 기준을 잡음
- `COALESCE(p.rating, 0) >= 3.5`
 - 최소 품질선을 지키기 위해서 평점 3.5 이상인 제품 필터링
- `COALESCE(r.reviewer_count, 0) >= 1`
 - 실제 리뷰어가 존재해야 신뢰성이 높아지기에 리뷰어 필터링

R (결과)

- 최소 평점 + 리뷰 확보 + 큰 할인율(내림차순 정렬) 상품 상위 20개 추출

```
SELECT
  p.product_id,
  p.product_name,
  p.actual_price,
  p.discounted_price,
  p.discount_pct,
  ROUND(p.price_ratio,2) as price_ratio,
  ROUND(COALESCE(p.bayesian_rating_small_m, p.rating),2) AS score_quality,
  p.rating,
  p.rating_count,
  r.reviewer_count
FROM `long-ceiling-470102-p4.amazon.products_analytics` p
LEFT JOIN `long-ceiling-470102-p4.amazon.product_reviewers` r USING(product_id)
WHERE p.discount_frac IS NOT NULL
  AND p.price_ratio BETWEEN 0 AND 0.80      -- 실가 ≤ 정가의 80%
  AND COALESCE(p.rating, 0) >= 3.5         -- 최소 품질선
  AND COALESCE(r.reviewer_count, 0) >= 1   -- 실제 리뷰어 존재
ORDER BY p.discount_frac DESC,
         score_quality DESC,
         COALESCE(r.reviewer_count,0) DESC
LIMIT 20;
```

쿼리 결과

결과 저장

다움에서 열기

작업 정보	결과	시작화	JSON	실행 세부정보	실행 그래프						
명	product_id		product_name	actual_price	discounted_price	discount_pct	price_ratio	score_quality	rating	rating_count	reviewer_count
1	8097C564GC		rt5 (2 Pack) Mini USB C Type C Adapter Plug, Type C Female to USB A Male Charger	4999.0	294.0	94.0	0.06	4.3	4.3	4426.0	8
			Charging Cable Adapter Converter compatible for								
2	80BF4YBLPX		Fire-Boltz Ninja Call Pro Plus 1.83" Smart Watch with Bluetooth Calling, AI Voice Assistance, 100 Sports Modes	19999.0	1799.0	91.0	0.09	4.2	4.2	13937.0	8
			IP67 Rating, 240*280 Pixel								
3	80BF5497ZT		Fire-Boltz Ninja Call Pro Plus 1.83" Smart Watch with Bluetooth Calling, AI Voice Assistance, 100 Sports Modes	19999.0	1799.0	91.0	0.09	4.2	4.2	13937.0	8
			IP67 Rating, 240*280 Pixel								
4	80BF563H84		Fire-Boltz Ninja Call Pro Plus 1.83" Smart Watch with	19999.0	1799.0	91.0	0.09	4.2	4.2	13937.0	8

페이지당 결과 수

50

1 - 20 (전체 20행)

2. 카테고리별 상위 5개

테마

카테고리 별 상위 5개의 적당한 품질과 가격대의 상품 추천

구현 로직

S (상황)

- 하나의 상품이 여러 카테고리 태그에 속하는 **멀티 라벨**
- 단순 인기순은 왜곡 가능성이 있어, 최소 품질선(평점/리뷰)과 가격 이상치 제거(price_ratio)가 필요
- 태그별로 공정한 비교를 위해 **태그 내에서만** 순위를 매겨야 함

T (과제)

- 모든 카테고리 태그에 대해 **신뢰성 있는 조건을 충족**하는 상품을 **태그별 상위 5개**로 안정적으로 선발
- 동점/동률 시에도 결과가 흔들리지 않도록 **결정적 정렬 규칙**을 적용

A (행동)

- **pc** CTE: `product_categories` 에서 `category` 를 **태그 단위로 정규화(TRIM)** 하여 `(product_id, category_tag)` 브리지 구성
- **scored** CTE: `products_analytics` 와 `pc` 를 `product_id` 로 **JOIN** → 각 상품에 **카테고리 태그**와 **품질/리뷰/할인** 지표를 결합
- **ranked** CTE:
 - **안전 필터**:
 - `COALESCE(rating, 0) >= 3.5` → 최소 품질선 확보
 - `COALESCE(review_count, 0) >= 1` → 실제 리뷰 확보
 - `price_ratio > 0 AND price_ratio <= 1` → 정가 대비 할인가 **이상치 제거**
 - **품질 스코어**: `COALESCE(bayesian_rating_small_m, rating)` → 소수 리뷰 과대평가를 억제
 - **태그별 순위 부여**:
 - `ROW_NUMBER() OVER (PARTITION BY category_tag ORDER BY 품질 DESC, review_count DESC, discount_frac DESC, product_id ASC)`
 - 품질 → 표본 크기 → 할인(가성비) 순으로 정렬, 마지막에 `product_id ASC` 로 **타이브레이크**
- 최종: `rk <= 5` 만 반환, `ORDER BY category_tag, rk` 로 태그별 Top-5를 깔끔히 정렬

R (결과)

- 각 카테고리 태그마다 **평점 3.5+ & 리뷰 ≥1**을 만족하고, **가격 이상치가 제거된 상위 5개** 상품을 제공
- 한 상품이 여러 태그에 속하면 **각 태그 랭킹에서 독립적으로** 노출

```
WITH pc AS (
  SELECT product_id, TRIM(category) AS category_tag
  FROM `long-ceiling-470102-p4.amazon.product_categories`
),
scored AS (
  SELECT p.*, pc.category_tag
  FROM `long-ceiling-470102-p4.amazon.products_analytics` p
  JOIN pc USING (product_id)
),
ranked AS (
  SELECT
    product_id,
    product_name,
    category_tag,
    COALESCE(bayesian_rating_small_m, rating) AS quality,
    review_count,
    discount_frac,
    ROW_NUMBER() OVER (
      PARTITION BY category_tag
      ORDER BY
        COALESCE(bayesian_rating_small_m, rating) DESC,
        review_count DESC,
        discount_frac DESC,
        product_id ASC
    ) AS rk
  FROM scored
  WHERE COALESCE(rating, 0) >= 3.5
    AND COALESCE(review_count, 0) >= 1
```

```

AND price_ratio > 0 AND price_ratio <= 1
)
SELECT *
FROM ranked
WHERE rk <= 5
ORDER BY category_tag, rk;

```

쿼리 결과

작업 정보

결과

시각화

JSON

실행 세부정보

실행 그래프

행	product_id	product_name	category_tag	quality	review_count	discount_frac	rk
1	B097JVLW3L	Irusa Play VR Plus Virtual Reality Headset with Headphones for Gaming (Black)	3DGlasses	3.504760503510...	8	0.23	1
2	B0978V2CP6	Cubetek 3 in 1 LCD Display V5.0 Bluetooth Transmitter Receiver, Bypass Audio Adapter with Aux, Optical, Dual Link Support for TV, Home	AVReceivers&Amplifiers	4.000535391903...	8	0.36	1
3	B08TGG316Z	10k 8k 4k HDMI Cable, Certified 48Gbps 10m Ultra High Speed HDMI 2.1 Cable 4k 120Hz 144Hz 2k 165Hz 8k 60Hz Dynamic HDR ARC eARC	Accessories	4.599313961160...	3	0.58	1
4	B09X7DVTQ4	SanDisk Extreme SD UHS-I 64GB Card for 4K Video for DSLR and Mirrorless Cameras	Accessories	4.499990163337...	8	0.48	2

페이지당 결과 수: 501 - 50 (전체 991명) < > >

3. 가성비 큐레이터

테마

메이저 베스트셀러 대신 **리뷰 수는 적지만(1~5개) 평점이 높은** 상품을 발굴합니다. 가성비 큐레이터 페르소나를 거냥해 **조용히 좋은 제품**을 표면화하고, 가격 비정상치는 배제해 신뢰도를 확보

구현 로직

S(상황)

- 전체적으로 리뷰 표본이 작아 단순 평점만 보면 과대평가 위험이 있음
- 그래도 **숨은 고품질 상품**을 발견해 보여주고 싶음

T(과제)

- 평점은 **높고(≥ 4.0)**, 리뷰 수는 **1~5개** 수준인 상품 중
- *가격 타당성($\text{price_ratio} \in (0,1]$)*을 만족하며 **품질 점수**가 높은 순으로 상위 20개 추천

A(행동)

- `products_analytics` (p)`와 `product_reviewers` (r)` 조인으로 평점/리뷰/가격 지표 결합
- 품질 점수는 `quality = COALESCE(bayesian_rating_small_m, rating)` 로 소수 리뷰 과대평가를 **완전히 억제**(스무딩 없으면 원평점 사용)
- 필터:
 - `COALESCE(p.rating, 0) >= 4.0` → 고품질
 - `COALESCE(r.review_count, 0) BETWEEN 1 AND 5` → 소수 리뷰
 - `p.price_ratio > 0 AND p.price_ratio <= 1` → 가격 이상치 제거
- 정렬: `quality DESC, p.discount_frac DESC` 로 **품질 우선**, 동률 시 **할인 많은 순**

R(결과)

- 리뷰는 적지만 **만족도가 높은** 상품을 상위 20개 반환
- 지금은 덜 알려졌지만 써본 사람들은 만족한 아이템을 빠르게 탐색 가능

```

SELECT
  p.product_id,
  p.product_name,
  COALESCE(p.bayesian_rating_small_m, p.rating) AS quality,
  p.rating,
  p.rating_count,
  r.reviewer_count,
  p.price_ratio
FROM `long-ceiling-470102-p4.amazon.products_analytics` p

```

```

LEFT JOIN `long-ceiling-470102-p4.amazon.product_reviewers` r USING(product_id)
WHERE COALESCE(p.rating, 0) >= 4.0
  AND COALESCE(r.review_count, 0) BETWEEN 1 AND 5
  AND p.price_ratio > 0 AND p.price_ratio <= 1
ORDER BY quality DESC, p.discount_frac DESC
LIMIT 20;

```

쿼리 결과

rank	product_id	product_name	quality	rating	rating_count	reviewer_count	price_ratio
1	B0BR4F878Q	Swiffer Instant Electric Water Heater Faucet Tap Home Kitchens Instantaneous Water Heater Tank less for Tap, LED Electric Head Water	4.799934636500...	4.8	53803.0	3	0.719859929954...
2	B08TGG316Z	10K 8k 4k HDMI Cable, Certified 48Gbps 1m Ultra High Speed HDMI 2.1 Cable 4k 120Hz 144Hz 2K 144Hz 8k 60Hz Dynamic HDR ARC eARC	4.599313961160...	4.6	3664.0	3	0.416423509795...
3	B09KXZQ0C1	Xiaomi Pad 5 Qualcomm Snapdragon 860, 120Hz Refresh Rate(6GB, 128GB), 2.5K+ Display (10.95-inch/27.81cm) 1 Billion	4.59912954053866	4.6	2886.0	4	0.710518697860...
4	B0BP7XLX48	Syncwire LTG to USB Cable for Fast Charging Compatible with	4.548292349725...	5.0	5.0	5	0.199599799899...

페이지당 결과 수: 50 1 - 20 (전체 20행) < > > >

4. 비슷한 가격대의 상위 대안

테마

같은 범주의 유사 가격대 중 품질(평점)이 더 좋은 대체 상품을 제안합니다. 지금 보는 제품과 비슷하지만 더 좋은 선택을 빠르게 보여주는 업셀/대체 제안 용 추천

구현 로직

S(상황)

- 사용자는 특정 상품을 보고 있음
- 같은 범주(쿼리에서는 `leaf_category`) 안에서 가격이 크게 벗어나지 않으면서 더 좋은 품질의 대안을 보고 싶어 함

T(과제)

- 대상 상품과 같은 범주이고 할인가가 $\pm 20\%$ 이내이며 품질이 같거나 더 높은 후보 중 상위 20개를 정렬해 제시

A(행동)

- `cur` CTE: 입력 `product_id` (예: 'B07XJWTYM2')의 가격/평점/품질 스코어(`quality = COALESCE(bayesian_rating_small_m, rating)`)와 범주(`leaf_category`)를 추출
- `pool` CTE: 전체 상품의 동일 지표들을 준비
- 본문:
 - 자기 자신 제외: `q.product_id != c.product_id`
 - 같은 범주만: `q.leaf_category = c.leaf_category`
 - 가격 근접: `q.discounted_price BETWEEN c.discounted_price * 0.8 AND c.discounted_price * 1.2`
 - 품질 우위/동급: `q.quality >= c.quality`
 - 정렬: 품질 내림차순 → 가격 차이(절대값) 오름차순으로 "더 좋고, 가격은 비슷한" 후보를 우선
- 안전장치 제안(옵션): `price_ratio ∈ [0,1]` 필터, `COALESCE` 로 결측 방어, 타이 브레이크(`product_id ASC`) 추가 등.

R(결과)

- 현재 상품과 같은 범주에서 더 좋거나 같은 품질이면서 가격이 비슷한 상위 20개 대체 후보를 반환
- 상세 페이지의 대체/업셀 영역에 그대로 노출 가능

```

WITH cur AS (
  SELECT
    p.product_id,
    p.product_name,
    p.discounted_price,
    p.rating,
    COALESCE(p.bayesian_rating_small_m, p.rating) AS quality,
    REGEXP_EXTRACT(cl.category, '([^\]]+)$') AS leaf_category

```

```

FROM `long-ceiling-470102-p4.amazon.products_analytics` p
JOIN `long-ceiling-470102-p4.amazon.catalog_light` cl USING(product_id)
WHERE p.product_id = 'B07XJWYM2'
),
pool AS (
SELECT
  p.product_id,
  p.product_name,
  p.discounted_price,
  COALESCE(p.bayesian_rating_small_m, p.rating) AS quality,
  REGEXP_EXTRACT(cl.category, r'([^\]]+$)') AS leaf_category
FROM `long-ceiling-470102-p4.amazon.products_analytics` p
JOIN `long-ceiling-470102-p4.amazon.catalog_light` cl USING(product_id)
)
SELECT
  q.product_id,
  q.product_name,
  q.discounted_price,
  q.quality,
  ABS(q.discounted_price - c.discounted_price) AS price_gap
FROM pool q
CROSS JOIN cur c
WHERE q.product_id != c.product_id
  AND q.leaf_category = c.leaf_category
  AND q.discounted_price BETWEEN c.discounted_price * 0.8 AND c.discounted_price * 1.2
  AND q.quality >= c.quality
ORDER BY q.quality DESC, price_gap ASC
LIMIT 20;

```

쿼리 결과

작업 정보	결과	시각화	JSON	실행 세부정보	실행 그래프
명	product_id	product_name	discounted_price	quality	price_gap
1	B08TV2P1N8	boAt Rockerz 255 Pro+ in-Ear Bluetooth Neckband with Up to 40 Hours Playback, ASAP Charge, IPX7, Dual Pairing, BT v5.0, with Mic (Active Black)	1399.0	4.099999879612...	280.0
2	B092N94QWQ	boAt Rockerz 330 in-Ear Bluetooth Neckband with Up to 35 Hours Playtime, ASAP Charge, Signature Sound, Dual Pairing & IPX5 with Mic (Active Black)	1499.0	4.099999844573...	180.0
3	B08BT5G5PM	Sennheiser CX 805 in-Ear Wired Headphones with in-line One Button Smart Remote with Microphone Black	1490.0	4.0999998262022	189.0

페이지당 결과 수: 50 1 ~ 3 (전체 3행) | < > > |

5. 예산 내 상위 품질 제품

테마

예산이 뻣뻣한 사용자를 위해, 최소 품질선을 지키면서 예산(≤ 200) 안에서 가장 가성비가 좋은 상품을 뽑아주는 리스트. 평점과 리뷰수(신뢰도), 그리고 할인을까지 동시에 고려해 싸고도 믿을 만한 후보를 추천

구현 로직

S(상황)

- 사용자는 예산 제약이 명확함(≤ 200)
- 단순히 싸기만 한 상품은 품질 문제가 있을 수 있으므로 평점/리뷰 기반 최소 품질선이 필요
- 할인을 표기가 다를 수 있어 가격 일관성(price_ratio) 검증도 요구됨

T(과제)

- 예산을 내 기본 품질(평점/리뷰)과 가격 타당성을 만족하는 상품 중, 가성비 점수 상위 20개를 선별

A(행동)

- 소스: `products_analytics` 에서 가격·평점·리뷰·할인 정보를 사용
- 하이브리드 점수(`score_hybrid`):

- $(COALESCE(rating, 3.0) + 0.05 * COALESCE(review_count, 0)) * (1 + 0.2 * COALESCE(discount_frac, 0))$
- **평점(품질) + 리뷰수(신뢰도)**를 합산하고, **할인 정도**에 비례해 소폭 가중
- **COALESCE**로 결측치 방어(평점 결측 시 3.0 가정, 리뷰수/할인 결측 시 0)
- 필터:
 - $discounted_price \leq 200.0$ → 예산 상한
 - $rating \geq 3.0$ & $review_count \geq 1$ → 최소 품질선/신뢰성
 - $price_ratio \in (0, 1]$ → 정가 대비 할인가 **이상치 제거**
- 정렬: **score_hybrid DESC**로 가성비 최적 후보 우선 노출

R(결과)

- 예산 내에서 **평점 3.0+ & 리뷰 ≥ 1**을 만족하고, **가성비 점수가 높은 상위 20개** 상품을 반환
- 평점·리뷰·할인 모두 반영되어 **현실적이고 설명 가능한** 리스트를 제공

```
WITH scored AS (
  SELECT
    p.*,
    -- 간단 하이브리드 점수: 평점 + 리뷰수 가중
    (COALESCE(p.rating, 3.0) + 0.05 * COALESCE(p.review_count, 0)) *
    (1 + COALESCE(p.discount_frac, 0) * 0.2) AS score_hybrid
  FROM 'long-ceiling-470102-p4.amazon.products_analytics' p
)
SELECT
  product_id, product_name, discounted_price, rating, review_count, discount_frac,
  score_hybrid
FROM scored
WHERE discounted_price IS NOT NULL
  AND discounted_price <= 200.0
  AND COALESCE(rating, 0) >= 3.0
  AND COALESCE(review_count, 0) >= 1
  AND price_ratio > 0 AND price_ratio <= 1
ORDER BY score_hybrid DESC
LIMIT 20;
```

쿼리 결과

입력 정보

결과

시각화

JSON

실행 세부정보

실행 그래프

명

product_id

product_name

discounted_price

rating

review_count

discount_frac

score_hybrid

1

B0BMAXML5MM

Lapster 65W compatible for OnePlus Dash Warp Charge Cable - type c to c cable fast charging Data Sync Cable Compatible with One Plus 10R

199.0

4.5

8

0.8

5.6840000000000...

2

B0BMAXML5MM

Lapster 65W compatible for OnePlus Dash Warp Charge Cable - type c to c cable fast charging Data Sync Cable Compatible with One Plus 10R

199.0

4.5

8

0.8

5.6840000000000...

3

B0BBSW21YC

LAPSTER 12pcs Spiral Cable Protectors for Charger Wires, Data Charger Cable Protector for Computers, Cell Phones etc (Grey)

99.0

4.4

8

0.9

5.6640000000000...

4

B00NH11PEY

AmazonBasics USB 2.0-A Male to A-Female Extension

199.0

4.5

8

0.73

5.6154

페이지당 결과 수: 50

1 - 20 (전체 20행)

<

>

>>