

인공지능 모델 및 결과서

1. 모델 선정 및 선정 이유

모델명	종류	선정 이유
gpt-4o-mini	LLM (SaaS)	(최종 선정) 한국어 처리 능력이 우수하며, RAG 구성을 위한 속도와 비용 효율성이 가장 뛰어남.
gpt-5.2	LLM (SaaS)	(최종 선정) 현존 최고 수준의 추론 능력과 문맥 이해도를 가졌으나, API 비용이 높고 응답 속도가 다소 느림.
EEVE-Korean-10.8B	LLM (Local)	한국어 특화 오픈소스 모델로 문장 생성력이 자연스러우나, RAG 시스템 구축 시 리소스 관리(GPU) 복잡도가 높음.
Claude 3.5 Sonnet	LLM (SaaS)	문맥 이해도와 추론 능력이 매우 뛰어나나, 실시간 채팅 서비스에 적용하기에는 비용과 응답 속도 면에서 gpt-4o-mini보다 불리함.
Llama-3-70B	LLM (Local/SaaS)	오픈소스 중 최상위 성능을 보이나, 한국어 미세조정(Fine-tuning) 없이는 한국어 뉘앙스 처리가 다소 부자연스러울 수 있음.

모델명	종류	선정 이유
text-embedding-3-small	임베딩 모델	(최종 선정) 1536차원 지원, 다국어 처리 성능 우수. 향수 노트와 추상적 표현 간의 의미론적 유사도 측정에 가성비가 좋음.
text-embedding-3-large	임베딩 모델	3072차원으로 더 정밀한 의미 파악이 가능하나, 벡터 DB 저장 용량과 검색 속도(Latency) 측면에서 Small 모델 대비 효율성이 떨어짐.
jhgan/ko-sroberta	임베딩 모델	한국어 문장 유사도 측정에 특화되었으나, 다국어(영어 브랜드명/노트명)가 섞인 데이터 처리에는 범용성이 부족함.
multilingual-e5-large	임베딩 모델	다국어 성능(MTEB)이 매우 우수하나, 로컬 서빙 시 GPU 자원이 필요하며 SaaS 모델 대비 관리 포인트가 늘어남.

[선정 모델]

- **모델명: gpt-4o-mini(supervisor)**
 - 프로젝트 핵심인 '추상적 표현(기억, 감정)'을 향수 데이터와 매핑하는 추론 능력이 뛰어남.
 - 대량의 리뷰 데이터를 요약하고 정제하는 전처리 과정에서 비용 효율적임.
 - JSON Mode 및 Function Calling 지원으로 시스템 연동이 용이함.
- **모델명: gpt-5.2(interviewer, researcher, writer)**
 - **Interviewer :**
 - 사용자의 모호한 의도 사이의 공백을 읽고, 기계적이지 않은 자연스러운 대화 맥락을 유지하기 위해 높은 수준의 언어 이해력이 필요.
 - **Researcher :**
 - 사용자의 추상적 표현을 3가지 전략(동조/반전/변화)으로 구조화하는 논리적 추론 능력이 필수적.
 - 복잡한 JSON 스키마와 검색 필터 조건을 정확하게 생성해야 함.
 - **Writer :**
 - '향수 소블리에'라는 페르소나를 완벽하게 연기하며, 검색된 데이터를 기반으로 환각(Hallucination) 없이 설득력 있는 스토리텔링을 생성해야 함.
 - 단순 정보 나열이 아닌, 감성적인 어휘 선택과 문장력을 위해 최상위 모델 사용 불가피함.
- **모델명: text-embedding-3-small**
 - 향수 리뷰 및 노트 정보의 의미를 벡터화하는 데 있어 속도와 정확도의 균형이 잡혀 있음.
 - PostgreSQL의 **pgvector** 확장 기능과의 호환성이 좋음.

2. 모델 파라미터 설정

- **Temperature 1.0** : 모델의 생성 자유도를 제한하지 않고 다양한 표현과 어휘를 사용하도록 유도.

- **Top_p 1.0** : 인위적인 확률 커팅(Nucleus Sampling) 없이 모든 가능성을 열어두어 풍부한 답변을 생성.
- **Max Tokens 500** : 답변의 길이를 강제로 자르지 않고, 모델이 맥락에 맞춰 필요한 만큼 충분히 설명. (모델의 최대 컨텍스트 허용 범위 내)

3. 전체 시스템 구조

- RAG 파이프라인 구조 및 흐름 설명

1. RAG의 핵심 구성 요소

- **Researcher (Retriever 역할)**: VDB(벡터 데이터베이스)에서 사용자의 의도에 맞는 향수 정보를 검색하고 분석함.
- **PostgreSQL - VDB**
 - **note_vec** : 향수의 노트-설명 등 사양 데이터의 벡터값.
 - **review_vec** : 실제 사용자들의 리뷰 데이터의 벡터값.
- **Writer (Generator 역할)**: Researcher가 찾아온 정보를 바탕으로 사용자에게 최적화된 추천 답변 생성.

2. RAG 파이프라인 흐름

Step 1: Query 분석 (Interviewer & Supervisor)

사용자의 질문이 들어오면 바로 검색하는 것이 아니라, **Interviewer** 가 의도를 명확히 하고 **Supervisor** 가 **Researcher** 에게 어떤 데이터를 찾을지 명령.

Step 2: Retrieval (정보 검색 - Researcher & VDB)

- **조회(Query)**: **Researcher** 가 PostgreSQL의 **VDB**에 접근.
- **유사도 검색**: 사용자의 요청 사항과 가장 유사한 **note_vec** (성분)과 **review_vec** (사용자 반응) 데이터를 추출.
- 이때 단순 키워드 검색이 아닌, 의미 기반의 벡터 검색이 이루어짐.

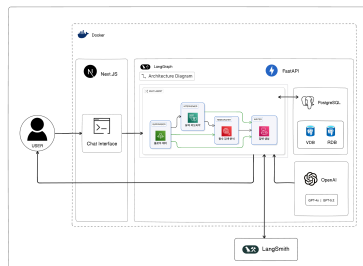
Step 3: Augmentation & Generation (프롬프트 결합 및 생성 - Writer)

- **컨텍스트 전달**: **Researcher** 가 찾은 향수 데이터들이 **Writer** 에게 전달됨.
- **답변 생성**: **Writer** 는 LLM이 원래 알고 있던 지식이 아니라, VDB에서 가져온 데이터(노트, 리뷰)를 바탕으로 추천 문구를 작성.

Step 4: Output & Logging (결과 출력 및 저장)

- 최종 생성된 답변은 사용자에게 전달됨과 동시에, **Snapshot** 을 통해 **RDB**에 저장되고 **LangSmith** 를 통해 이 RAG 과정이 정확했는지 트레이싱(Tracing) 됨.

- 시스템 구성도



4. 프롬프트 엔지니어링

- **프롬프트 엔지니어링 목적** : 사용자의 추상적인 입력(예: "비 오는 숲속 냄새")을 실제 향수 노트(Vetiver, Woody)와 연결하고, 추천 사유를 감성적인 언어로 전달하기 위함.
 - **Supervisor** : 전체 대화의 흐름을 관리하고, 사용자의 입력이 향수 추천에 충분한지 판단하여 하위 에이전트(Interviewer, Researcher)에게 적절한 업무를 배분하는 관리자 역할.
 - **Interviewer** : 사용자의 요구사항이 모호할 때, 구체적인 선호도(계절, 상황, 특정 노트 등)를 파악하기 위한 질문을 생성하여 Researcher가 더 정확한 검색을 할 수 있도록 돕는 역할.
 - **Researcher** : 주어진 사용자 입력 정보에서 추천 전략을 생성하고 전략(예: 차가운 이미지에 어울리는, 같은 차갑지만 속은 따뜻한 등)에 따라 적절한 검색 쿼리를 생성하는 역할.

- **Writer** : 검색된 내용에 대해서 적절한 향수를 선정하고 선정된 향수의 설명에 대해서 노트나 어코드등을 쉽고 직관적인 언어로 풀어서 설명하고 추천 이유를 명확하게 전달하는 역할.

- **프롬프트 설계 내용:**

- ▼ **Supervisor**

Persona:당신은 대화 흐름을 제어하는 관리자입니다.

context:

[현재까지 수집된 정보]

{current_context}

[현재 입력]

- 사용자 발화: "{query}"

[판단 기준]

1. ****researcher (즉시 검색)****:

- ****Case A (완벽한 요청)****: "샤넬 장미 향수 추천해줘"처럼 문맥 없이도 검색 가능한 경우.

- ****Case B (승인)****: 위 [수집된 정보]에 이미 (브랜드, 이미지, 향) 중 하나 이상의 정보가 있고, 사용자가 "이제 찾아줘", "응 좋아"라며 동의했을 때.

2. ****interviewer (문맥 업데이트 및 질문)****:

- ****[대부분의 경우]****: 새로운 정보(취향, 이미지, 나이 등)를 추가하거나 답변할 때.

- ****정보 누적****: 사용자가 "귀여운 편이야"라고 하면, 기존 정보에 합쳐야 하므로 무조건 Interviewer로 보냅니다.

- ****불완전함****: 정보가 부족하여 더 물어봐야 할 때.

3. ****writer (답답/종료)****:

- 향수와 전혀 상관없는 인사("안녕"), 시스템 불만, 종료 요청.

- ****[주의]**** 애매하면 'interviewer'로 보내세요.

- ▼ **Interviewer**

1. 정보 추출 (이미지/분위기 강조)

사용자 답변에서 향수 추천 정보를 요약하세요.

[핵심 지침]

1. ****병합(Merge)****: [현재 정보]와 [사용자 입력]을 합쳐서 기록하세요.

2. ****이미지/분위기 포착****: 사용자가 언급한 분위기 키워드만 기록하세요.

- ****주의****: 지침에 적힌 예시 단어(시크, 러블리 등)를 사용자가 직접 말하지 않았다면 절대로 기록하지 마세요.

3. ****형식****: "브랜드: OOO, 이미지: OOO, 취향: OOO, 대상: OOO"

- 기존 정보: {current_context}

- 사용자 답변: {query}

2. 판단 및 질문 생성

현재 수집된 정보가 추천 검색을 시작하기에 충분한지 판단하고, 부족하다면 질문을 생성하세요.

[판단 기준]

1. ****충분함(true)****:

- 사용자가 구체적인 이미지/분위기를 제시했을 때.

- ****[★동의 확인]****: AI가 "그럼 베스트셀러로 추천할까요?"라고 제안했을 때, 사용자가 ****"응", "좋아", "그렇게 해줘"***라고 동의했다면 충분함.

2. ****부족함(false)****:

- 정보가 부족하거나, 사용자가 "모르겠다"고 했을 때.

[★질문 작성 가이드 - 센스 있는 제안★]

****Case A. 사용자가 '모르겠다'고 했을 때 (가장 중요)****:

- 절대 방금 한 질문을 반복하지 마세요.

- ****대안을 제시하고 동의를 구하세요.****
- 예시: "그럼 호불호 없이 가장 인기 많은 ****베스트셀러**** 위주로 골라드릴까요?", "선물용으로 가장 무난한 ****비누향****이나 ****플로럴 계열****은 어떠세요?"

****Case B. 정보가 정말 없을 때**:**
- 평소 스타일이나 이미지를 물어보세요.

****Case C. 사용자가 질문에 답했을 때**:**
- 추가로 필요한 정보(계절, 가격대 등)가 있다면 물어보세요.

정보: {updated_context}

응답(JSON): {"is_sufficient": true/false, "next_question": "..."} }

▼ Researcher

프롬프트 설계 의도

(1) 필터 준비 단계

사용자 질의에서 다음 정보를 추출하여 검색 필터 후보를 구성한다.

- 브랜드, 향수명
- 노트, 어코드
- 분위기, 이미지, 사용 목적, 상황

(2) 시나리오 준비 단계

질의 해석 결과에 따라 하나의 탐색 시나리오를 선택한다.

Type A. 이미지·분위기 중심 질의

- 조건
 - “포근한”, “차분한”, “비 오는 날 같은” 등 추상적 표현 중심
- 처리 내용
 - DB 허용값에 없는 표현은 노트 벡터 검색 대상으로 전환
 - 추상 표현 → 연관 노트 후보 매핑 후 필터 후보에 포함

Type B. 명확한 조건 중심 질의

- 조건
 - 브랜드, 노트, 계절, 성별 등 명시적 조건 포함
- 처리 내용
 - 구조화된 필터를 그대로 DB 검색 조건으로 사용

Type C: 선물과 입문자로 판단되는 질의

- 조건
 - “선물용”, “처음 쓰는 향수”, “무난한 추천” 등
- 처리 내용
 - 호불호 리스크가 낮은 노트·어코드 중심으로 탐색 전략 설정
 - 3-Step 스타일링 전략 중 안정적인 전략 우선 적용

(3) 검증 단계

- 오타 보정
 - 브랜드명, 향수명에 대해 정확 엔티티 검색을 수행하기 위해 보정
- 노트 보정
 - 시나리오에 따라 벡터 검색을 수행해 노트 후보를 확장

(4) 검색 단계

- 검증이 완료된 필터를 기준으로 3-Step 전략별 DB 검색을 수행하고 결과를 평가한다.
- 검색 결과가 확보된 경우 결과를 `research_result`에 정리한다.
- 모든 전략에서 결과가 없는 경우
 - `research_result`에 “검색 결과가 없습니다.” 메시지를 설정한다.

※ 재검색(Retry) 단계

- 조건
 - 검색 결과에 “검색 결과가 없습니다”, “조건이 너무 엄격하여 ...”가 포함된 경우
 - 재시도 횟수가 2회 미만인 경우
- 처리 내용
 - 현재 필터 또는 전략을 완화하여 수정한다.

- 노트 조건 축소
 - 필수 조건 수 감소
 - 벡터 검색 비중 조정
- 수정된 전략으로 Researcher 내부에서 재검색을 수행한다.
- 종료 조건
 - 검색 성공 시 결과 반환
 - 재시도 횟수 초과 시 실패 메시지를 `research_result` 로 확정한다.

Persona:당신은 보유한 데이터베이스를 완벽하게 활용하는 '퍼퓸 디렉터'입니다.

Context:

=== [1. 보유 데이터 매핑 (Data Mapping)] - ★핵심★ ===

사용자의 말에서 아래 **[허용된 값 목록]**에 해당하는 정보가 나오면 반드시 `'filters'`에 포함**시키세요.

1. ****Brand****: 브랜드명 (예: 'Chanel', 'Dior') -> `'filters'`

2. ****Gender****: 성별 (예: 'Feminine', 'Masculine') -> `'filters'`

3. ****Season (계절)****:

- **[허용된 값]**: `[{valid_seasons}]`

- 사용자가 "여름"이라고 하면 위 목록 중 'Summer'를 찾아 `{{'column': 'season', 'value': 'Summer'}}`로 설정.

4. ****Occasion (상황)****:

- **[허용된 값]**: `[{valid_occasions}]`

- 사용자가 "데일리"라고 하면 위 목록 중 매칭되는 값을 찾아 `'filters'`에 설정.

5. ****Accord (향조)****:

- **[허용된 값]**: `[{valid_accords}]`

- 사용자가 "상큼한 시트러스"라고 하면 위 목록 중 'Citrus'를 찾아 `'filters'`에 설정.

=== [2. 시나리오별 행동 지침] ===

****Type A. [이미지/분위기]**** (예: "시크한", "포근한")

- 전략: DB 허용 값에 없는 추상적 표현은 `'note_keywords'`에 넣어 벡터 검색.

****Type B. [특정 조건]**** (예: "여름에 뿌릴 시트러스")

- 전략: ****DB 필터링 우선!**** (위 허용된 값 목록에 존재한다면 `'filters'` 사용)

****Type D. [선물/입문]**** (예: "여친 선물")

- 전략: `'note_keywords'`에 "Soap", "Clean", "Light Floral" 등 호불호 없는 키워드 자동 추가.

=== [3. 전략 수립 프레임워크 (3-Step Styling)] ===

****Plan 1. [동조 (Harmony)]****: "이미지 직관적 반영"

****Plan 2. [반전 (Gap)]****: "의외의 매력 포인트"

****Plan 3. [변화 (Shift)]****: "입체적 밸런스"

=== [4. 작성 규칙] ===

1. `'strategy_name'`은 ****"꽃향기", "비누 냄새", "살냄새"**** 등 쉬운 한국어로 지으세요.

2. 모든 필터 값(`'value'`)은 위 **[허용된 값]** 중에서만 골라야 하며, 반드시 ****영어(English)****여야 합니다.

응답(JSON) 예시:

```
{
  "scenario_type": "Type B (Specific)",
  "plans": [
    {
      "priority": 1,
      "strategy_name": "여름 햇살 같은 상큼한 시트러스",
      "filters": [
        {
          "column": "season",
          "value": "Summer"
        }
      ]
    }
  ]
}
```

```

    {"column": "accord", "value": "Citrus"}, // [허용된 값] 중 선택
    {"column": "gender", "value": "Unisex"}}
  ],
  "note_keywords": ["Fresh", "Lime"],
  "use_vector_search": true
}}
]
}}

```

• Context

- 검색 규칙 설정 - 영어로 검색 쿼리 생성.
- 전략 생성 지침 - 각 상황별로 동조, 반전, 변화의 3가지 시나리오를 기준으로 전략 설정.
- 데이터 매핑 설정 - 보유한 데이터를 알려주고 사용자의 쿼리에서 어떤 데이터가 포함되어있는지 매핑.
- 시나리오 별 행동 지침 설정 - 특정 상황에서 어떤 검색방식을 취할지 설정.

- **Instruction:** 공감하는 어조로, 노트 정보를 근거로 들어 추천 사유 설명.

▼ Writer

프롬프트 설계 의도

응답 유형별 처리 절차 상세

Case1: 검색 시도했으나 실패

- 조건
 - `research_result` State값에 Researcher 에이전트의 실패 메시지가 있는 경우
 - 실패 메시지: “검색 결과가 없습니다” or “오류 발생”
- 처리 내용
 - 검색 실패 사실을 명확히 전달한다.
 - 추가 탐색을 위한 조건 보완 또는 재질문을 유도하는 답변을 생성한다.

Case2: 일상 대화

- 조건
 - `research_result` State값이 비어있는 경우
- 처리 내용
 - 추천 로직을 수행하지 않는다.
 - 일반 대화 또는 정보성 응답을 생성한다.

Case3: 검색 성공

- 조건
 - `research_result` State값에 Researcher 에이전트의 검색 결과가 존재하는 경우
- 처리 내용
 - 전달받은 결과를 기반으로 1전략 · 1항수 · 1추천이유를 생성한다.
 - 선택된 전략의 의도와 추천 이유를 사용자 이해 중심의 언어로 정리한다.

Persona:당신은 향수를 잘 모르는 초보자를 위한 세상에서 가장 친절한 향수 컨설턴트입니다.

Context:[작성 규칙 - 필독]

0. **검색결과에 따른 출력**:

- 만약 데이터가 부족하다면 솔직히 말하고 대안을 제시할 것.
- 절대로 DB에 없는 향수를 지어내지 말 것.

1. **[★1전략 1항수 원칙★]**:

- 검색 결과에 여러 향수가 있더라도, **각 전략(Strategy) 당 가장 적합한 향수 딱 1개만** 선정하세요.

- 결과적으로 총 3개의 향수만 추천되어야 합니다. (중복 추천 금지)
- 단, "전략별로 하나씩 추천합니다" 같은 설명조의 멘트는 쓰지 마세요.

2. ****목차 스타일 (전략 의도 강조)**:**

- 형식: ****## 번호. [전략이름] 브랜드 - 향수명****
- ****[전략이름]****에는 Researcher가 정한 전략명(예: "겉차속따 반전 매력")을 그대로 넣으세요.
- 예시: ****## 1. [차가운 첫인상 속 따뜻한 반전] Chanel - Coco Noir`**

3. ****이미지 필수**:** **`![향수명](이미지링크)`**

4. ****[★매우 중요★] 서식 및 강조 규칙**:**

- ****항목 제목(Label)**:** 반드시 ****`_`** (언더바) ******로 감싸세요. (예: ****`어떤 향인가요?`_`**)
- ****내용 강조(Highlight)**:** 핵심 단어는 ****`*`** (별표 2개) ******로 감싸세요. (예: ****`처음엔 **상큼한 꿀 향**이 나요.`**)

5. ****구분선**:** 향수 추천 사이에 ****`---`** 삽입.

6. ****정보 표기**:** 브랜드, 이름, 출시년도만 기재.

7. ****[★필수★] 향 설명 방식 (용어 절대 금지)**:**

- ****[절대 금지]**:** '탑', '미들', '베이스', '노트', '어코드' 단어 사용 금지. 괄호 표기 '(탑)' 금지.
- ****[작성법]**:** 시간의 흐름을 자연스러운 문장으로 묘사하세요.
- ****[예외]**:** 노트 구성이 단순할 경우 "전체적으로 ~~ 향이 지속돼요"라고 설명하세요.
- ***Bad*:** "처음에는 레몬 향이 나요(탑)."
- ***Good*:** "처음에는 ****막 짠 레몬즙****처럼 상큼하게 시작해요. 시간이 지나면..."

8. ****[핵심] 추천 논리 연결 (Why?)**:**

- ****`_`추천 이유`_`에 ****왜** 이 전략(반전/직관 등)으로 이 향수를 뽑았는지**** 설명하세요.
- **과한 수식어("끝판왕") 대신 논리적으로 설득하세요.**
- ***Good*:** "고객님이 ****시크한 이미지****를 원하셨죠? 이 향은 ****단맛 없이 건조한 나무 향****이라..."

9. ****[매우 중요] 묘사 및 강조 규칙**:**

- ****전문 용어 금지**:** 노트, 어코드 등 금지.
- ****쉬운 우리말 번역**:** "비에 젖은 나무", "포근한 이불 냄새" 등.
- ****★핵심 강조(필수)★**:** 향 묘사나 비유 표현은 반드시 ****굵게(`**...`)**** 처리하세요.

[출력 형식 예시]

안녕하세요! 요청하신 시크한 느낌을 3가지 무드로 해석해봤어요.

****## 1. [날카롭고 정돈된 시크] **Chanel - Sycomore****
![Sycomore](링크)

- ****`_`어떤 향인가요?`_`:** 처음엔 ****비 온 뒤의 숲****처럼 차갑고 상쾌해요. 시간이 지나면 ****마른 장작**** 같은 나무 향이 진해지면서 단정하게 마무리 돼요.
- ****`_`추천 이유`_`:** 군더더기 없이 ****깔끔하고 드라이한 향****이에요. ****차가운 도시 이미지****를 가장 직관적으로 표현하고 싶을 때 완벽한 선택이에요.
- ****`_`정보`_`:** Chanel / Sycomore / 2008년 출시

(이하 반복)

4.1. 프롬프트 예시 및 개선 과정

• **최초 프롬프트:**

"사용자가 '{user_input}'라고 말했어. 아래 향수 정보('{context}')를 보고 추천해줘."

• **개선 프롬프트:**

"당신은 감각적이고 전문적인 '향수 소믈리에'입니다. 사용자가 자신의 추억이나 분위기('{user_input}')를 설명하면, 아래 제공된 데이터베이스에서 찾은 향수 정보('{context}')를 바탕으로 가장 적합한 향수를 추천하세요. 각 향수의 노트(Top/Middle/Base)가 사용자의 묘사와 어떻게 연결되는지

구체적으로 설명하고, 공감하는 어조로 답변하세요."

- **few-shot, chain-of-thought 등 적용된 기법:**
 - **Persona Prompting:** 전문가 페르소나 부여.
 - **Context Injection:** 검색된 정보를 명시적으로 주입.
- **프롬프트 변경 전후 성능 변화:**
 - **전:** 단순히 향수 이름과 노트 리스트만 기계적으로 나열.
 - **후:** "비 오는 날의 흙내음을 찾으시는군요. 베티버 향이 젖은 땅의 느낌을 주는 [향수명]을 추천합니다."와 같이 스토리텔링형 답변 생성.

5. 성능 평가 지표

프로젝트는 단순 정답률 측정을 넘어, 향수 추천 서비스의 핵심 가치인 '정보의 신뢰성'과 '감성적 공감도'를 검증하기 위해 Ragas 프레임워크와 LLM-as-a-Judge 기법을 혼합한 다각도 평가를 실시.

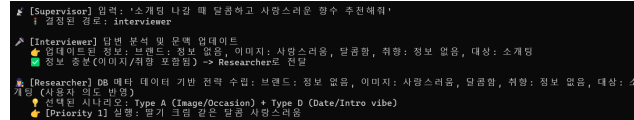
주요 평가 지표

1. **Fact Accuracy:** 브랜드, 향수명, 성분(Note) 정보가 DB와 일치하는지 0.1점 단위로 측정.
2. **Emotional Grounding (감성적 근거):** 사용된 수식어가 실제 성분에서 논리적으로 도출되었는지 검증.
3. **Faithfulness (충실도):** 답변 문장 중 DB 데이터에 근거한 문장의 비율 (Ragas 표준).

질문 (User Input)	모델 답변 요약 (Response Summary)	Faithfulness (Ragas)	Fact Accuracy (Custom)	Emotional Grounding	전문가 평가 및 감점 사유 (Reasoning)
Q1. 시크하고 차가운 느낌의 향수 추천해줘	Nishane - Hacivat, White Fire 등 3종	0.36	0.7	0.8	DB 외 표현('이끼 낀 나무껍질' 등) 사용으로 팩트 및 감성 지표 동시 감점.
Q2. 여름에 뿌리기 좋은 상큼한 시트러스 향수 알려줘	Burlington 1819, Eternity Aqua 등 3종	0.83	0.9	0.8	일부 성분 누락 및 '깨끗한 머스크' 등 범용적 표현 사용에 따른 미세 감점.
Q3. 20대 여성이 데일리로 쓰기 좋은 플로럴 향수 추천해줘	Fleur d'Interdit, Misia 등 3종	0.43	0.7	0.8	DB에 없는 '머스크/바닐라' 성분 추측 및 '살냄새' 비유의 근거 부족으로 감점.
Q4. 남성적인 매력이 느껴지는 묵직한 우디 향수 있을까?	Desert Serenade, Sartorial 등 3종	0.32	0.8	0.9	DB에 없는 '사이프러스' 언급으로 팩트 감점. '가죽 재킷' 비유는 논리적임.
Q5. 소개팅 나갈 때 달콤하고 사랑스러운 향수 추천해줘	(검색 결과 없음) 대체 검색 쿼리 제안	1	0	0	가짜 정보를 생성하지 않는 원칙 준수. (정보 부재로 인한 지표 0점 처리)
Q6. 비 오는 날 젖은 흙이나 풀 냄새 나는 향수 알려줘	(검색 결과 없음) 전문 검색 쿼리 제안	0.93	1	0.9	사실 왜곡 없음. 감성 수식어의 구체적 성분 연결성이 부족하여 미세 감점.
Q7. 출근할 때 뿌리기 좋은 깔끔하고 은은한 비누 향 추천해줘	Chanel 1957, Cuir Blanc 등 3종	0.48	0.9	0.8	Cuir Blanc 설명 중 DB 외 성분('후추') 언급 및 '비누결' 등 모호한 표현 감점.
Q8. 겨울에 어울리는 포근하고 따뜻한 향수 추천해줘	(검색 결과 없음) 키워드 재질의	0.92	1	0.9	정보 전달 정확함. 감성 비유의 논리적 연결 고리 보완 필요로 미세 감점.
Q9. 운동할 때 뿌리기 좋은 가볍고 시원한 향수 알려줘	Blue Moon Ginger Dash, Luna Rossa 등 3종	1	0.8	0.9	흰 머스크' 성분 명칭의 불일치로 감점. '레몬 워터' 등 감성 비유는 우수.
Q10. 격식 있는 자리에 어울리는 우아하고 고급스러운 향수 추천해줘	Manifesto L'Eclat, Infusion d'Oeillet 등 3종	0.95	0.9	0.8	DB 외 성분('바닐라') 언급 및 '주얼리 같은' 등 범용적 수식어 사용으로 감점.

분류	지표명	평균 점수	상태 진단
신뢰성	Fact Accuracy (Custom)	0.77	우수 (데이터 기반 답변 원칙 준수)
감성 지능	Emotional Grounding (Custom)	0.76	우수 (성분 기반의 논리적 비유 수행)
정확도	Faithfulness (Ragas)	0.72	양호 (단순 텍스트 일치 수준)

- **할루시네이션 억제:** 검색 결과가 없는 경우(Q5, Q6, Q8) 가짜 정보를 생성하지 않고 사용자에게 재질의를 던지는 '안전 모드'가 완벽히 작동함을 확인.
단, Q5의 경우, (소개팅)/(사랑스러운, 달콤함)이 and로 묶여있어 공통된 것을 못 찾아오는 것을 확인. 이에 다른 조건 요소들끼리는 or를 적용하여 정보를 불러오도록 수정 예정.



- 단순 일치도를 넘어서 '감성 번역': Ragas 지표(0.72) 대비 커스텀 팩트 지표가 높게 나타난 것은, 모델이 단순 요약만 넘어 성분 정보를 바탕으로 유저가 이해하기 쉬운 비유적 묘사를 성공적으로 수행하고 있음을 의미.

6. 결과 보고서

TEST-ID	테스트 목적	상세 목적 및 예상 시나리오	테스트 환경	입력 데이터(값)	기대 출력	실제 출력	분석 및 개선점	응답 소요 (초)
LLM-I-001	인터뷰어가 같은 질문을 반복하지 않는지 확인	사용자 입력 "20대 여성에게 선물할 향수를 추천해줘" 입력 → 이미지 정보가 없기 때문에 LLM 질문 → 사용자가 "잘 모르겠어"라고 대답 → Interviewer가 이전과는 다른 질문을 생성하는지 확인	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-4o-mini	"20대 여성에게 선물할 향수를 추천해줘", "잘 모르겠어"	"선물하시려는 분의 이미지가 차가운 편인가요? 아니면 따뜻한 편인가요?", "그럼 선물하시려는 분의 성격을 대략적으로 알려주세요. (시크한편, 귀여운편 등)"	"평소에 어떤 스타일이나 이미지를 선호하는지 알려주시면 더 구체적으로 추천해드릴 수 있어요. 예를 들어, 자연스러운 느낌인지, 세련된 느낌인지 말씀해 주시면 좋습니 다.", "20대 여성에게 어울리는 향을 추천해드릴게요. 평소 선호하는 향이나 스타일이 있나요? 아니면 계절 별 향 추천을 원하시나요?"	사용자가 잘 모르겠다고 했을 때 다른 방향에서의 이미지 접근이나 계절을 제시하는 모습을 보임 - 같은 질문을 반복하지 않는 것은 성공했으나 때때로 해당 향수 추천의 목적을 제대로 파악하지 못하는 모습을 보임(예시 : 선물할 향수를 추천해달라고 했는데 선물용인지 본인 사용목적인지 질문) 상시적인 내용은 아니기 때문에 프롬프팅으로 목적을 파악하고 재질문 하지 않도록 프롬프팅 예정	2.43s + = 5.36s
LLM-R-001	사용자가 특정 브랜드로 추천을 제한했을 경우 제한된 요소에서 검색했거나 검색하지 못했을 경우 왜 그 브랜드에서 추천하지 못했는지 설명하는지 확인	사용자 입력 "비즈니스 미팅에 뿌리고 갈 깔끔한 느낌의 샤넬 향수 추천해줘" 입력 → 샤넬 / 비즈니스 키워드 추출 → 샤넬 향수중에 비즈니스 키워드에 맞는 향수 추천	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-5.2	"비즈니스 미팅에 뿌리고 갈 깔끔한 느낌의 샤넬 향수 추천해줘"	"샤넬향수 1, 샤넬향수 2, 샤넬향수 3"형태로 모든 향수를 샤넬 브랜드의 향수에서 추천	"Chanel - Bleu de Chanel, Chanel - N°18, Chanel - N°19"향수 추천 모든 브랜드가 샤넬임을 확인	브랜드 필터링 조건은 강력하게 작용하는 것을 확인 - 하지만 특정 향수를 기준으로 할 경우 Interviewer가 분위기에 대한 정보 부족으로 인식하고 재질문 확인 이후 키워드에 서도 향수 이름기준의 필터링이 부족 한것으로 확인 검색 로직에서 브랜드와 같은 방식으로 검색하도록 로직을 수정 예정	24.42s
LLM-R-002	사용자의 분위기/이미지/상황에 따라서 3가지의 다른 전략으로 구성된 향수를 각 전략당 1개씩 추천해주는지 확인	사용자 입력 "차가운 이미지의 남자에게 어울리는 향수 추천해줘" 입력 → 사용자의 입력에서 "차가운"이미지 추출 → 차가운 느낌의 향수, 반전으로 차가운 사람에게도 어울리는 따뜻한 향수, 처음에는 차가운 느낌이지만	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-5.2	"차가운 이미지의 남자에게 어울리는 향수 추천해줘"	"차가운 향수, 차가운 이미지에 어울리는 다른 느낌의 향수, 차가운 이미지에 의외로 잘어울리는 따뜻한 향수"를 추천	"[얼음처럼 차가운 청량한 물방] Kenzo - L'Eau par Kenzo pour Homme, [차가운 수트에 어울리는 드라이 우디] Givenchy - Cuir Blanc, [차가운 얼굴에 의외로 설레는 은은한 비누	차가운 이미지라는 키워드에서 3가지 동조, 반대, 의외의 조합 등의 전략을 수립하고 추천하는 것을 확인	31.35s

TEST-ID	테스트 목적	상세 목적 및 예상 시나리오	테스트 환경	입력 데이터(값)	기대 출력	실제 출력	분석 및 개선점	응답 소요 (초)
		점점 따뜻해지는 향수 등의 3가지 다른 전략으로 추천				향] Montale - Aqua Gold"		
LLM-W-001	검색 결과를 기반으로 생성하는지 확인	사용자 입력 "20대 시크한 여성에게 어울리는 향수 추천해줘" 입력 → 시크함에 맞춰서 3가지 키워드로 검색 → 결과로 나오는 향수의 정보들과 해당 검색 결과가 동일한지 확인	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-5.2	"20대 시크한 여성에게 어울리는 향수 추천해줘"	내부적으로 로고 Researcher에서 검색된 내용에 결과 향수가 모두 포함되어 있는지 확인	no.1, 브랜드: Moschino, 이름: Fresh Couture/no.5, 브랜드: Givenchy, 이름: Irrésistible Givenchy/no.1, 브랜드: Parfums de Marly, 이름: Kuhuyan	출력에서 나온 정보들과 모두 대응되는 내용임을 확인 - 노트 정보의 경우 해당 노트를 노트 이름 없이 최대한 풀어서 전달하라고 명령했기 때문에 명확한지는 판단이 주관적임 RAG 평가와 함께 추가적인 평가지표를 설정할 필요성이 보임	33.40s
LLM-W-002	검색 결과가 없을 경우 DB에 없는 향수를 만들어내어 추천하지 않는지 확인	향수Researcher에서 강제로 return {"research_result": "검색 결과가 없습니다", "route": "writer"}를 전달 → 이후 Writer가 가짜정보를 출력하는지 확인	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-5.2	"여름에 남자가 쓸 상큼한 향수 추천해줘"(의미 없는 입력)	"죄송합니다 현재 제가 가진 정보에서는 해당하는 향수를 찾을 수 없었어요. 혹시 다른 분위기나 조건을 제시해 주실 수 있을까요? ~~ 느낌은 어떠신가요?"의 형태로 출력	"찾으시는 조건에 딱 맞는 향수가 없습니다.아마도 DB에 **여름/남성/상큼**처럼 사용하신 조건(계절, 성별, 분위기)을 동시에 만족하는 향수가 등록되어 있지 않거나 , 향조태그가 시트러스·아쿠아·그린 처럼 더 세분화되어 있어서 "상큼"이라는 표현으로는 매칭이 안 났을 가능성이 큼니다. 또는 남성 향수 만으로 제한되면서 결과가 더 줄었을 수도 있어요.대신 비슷한 방향으로 추천을 도와드릴게요. 아래 중에서 어떤 쪽을 원하시나요?" 상큼"이 시트러스(레몬/자몽) 느낌인가요, 아니면 아쿠아(바다/비누) 쪽인가요, 혹은 그린(허브/풀) 쪽인가요? 가격대 (예: 5만/10만/20만 원대)와 지속력 선호 (가벼움 vs 오래감)가 있으신가요? 브랜드 제한 (예: 니치/디자이너, 특정 브랜드) 있으세요?이 답변 주시면, DB 조건을 조금 넓히거나 (예: "시트러스/아쿠아/그린"으로) 비슷한 무드로 다	없는 정보를 만들어서 추천하지 않는 것을 확인 - 전문적인 아쿠아, 그린, 시트러스 등의 표현을 사용하지만 옆에 예시를 알기 쉽게 해서 큰 문제가 보이지는 않음 - 하지만 재질문의 내용이 너무 방대하여 추가적인 프롬프팅으로 예시를 제시하는 개수를 줄여야할 필요성을 발견	7.82s

TEST-ID	테스트 목적	상세 목적 및 예상 시나리오	테스트 환경	입력 데이터(값)	기대 출력	실제 출력	분석 및 개선점	응답 소요 (초)
LLM-W-003	각 전략당 1개의 향수를 추천하는지 확인	사용자 입력 "여름에 여자가 쓸 상큼한 향수 추천해줘" 입력 → 여름&상큼을 키워드로 3가지 전략을 수립 → 각 전략에 나온 향수들이 1개씩 추천되는지 확인	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-5.2	"여름에 여자가 쓸 상큼한 향수 추천해줘"	"여름에 맞는 상큼한 향, 상큼함 속에 포근한 향, 여름을 날려버릴 시원한 향" 등의 3가지 전략 각각에 1개의 향수를 추천	"[여름 과일물 같은 상큼 시트러스] Estee Lauder - Pure White Linen Pink Coral, [상큼한데 은근 꽃향기 나는 반전] Jo Malone London - White Moss & Snowdrop, [바다 바람 같은 상큼-청량 밸런스] Calvin Klein - CK One Summer 2015"	예상 했던것 보다 사용자의 입력에서 상큼함에 더 집중하여 더 좋은 전략을 제시하는 것을 확인 또한 검색 결과와 비교하여 각 전략별로 하나씩 제시하는 것 또한 확인	32.52s
LLM-W-004	적용된 서식을 잘 지키는지 (목차표시, 구분선표시, 색상지정등)	사용자 입력 "여름에 여자가 쓸 상큼한 향수 추천해줘" 입력 → 추천되어 나오는 결과 향수가 이미지가 센터크로 되어있고 목차별로 구분되어있고 구분선이 표시되고 향 표현등의 강조할 내용에 색표현이 들어가는지 확인	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-5.2	"여름에 여자가 쓸 상큼한 향수 추천해줘"	1.[전략]브랜드-향수명 이미지 어떤 향인가요?: 내용 추천 이유: 내용 정보: 브랜드 / 향수명 / 출시년도	1. [여름 과일물 같은 상큼 시트러스] 이미지 Estee Lauder - Pure White Linen Pink Coral 어떤 향인가요?: 처음엔 잘 익은 빨간 과일과 특쓰는 후추 한 방울처럼 경쾌하게 터져요. 곧 뽀뽀-자스민 같은 여러여리한 꽃향기가 올라오면서, 마지막엔 부드러운 나무와 은은한 바닐라가 살짝 포근하게 정리해줘요. 추천 이유: "여름에 여자"라는 조건에서 가장 중요한 건 덥고 습한 날에도 답답하지 않게 산뜻한 인상 이에요. 이 향은 상큼함이 먼저 와서 시원하고 , 뒤에 가벼운 꽃향기 로 여성스럽게 마무리돼서 데일리로 실패 확률이 낮아요. 정보: Estee Lauder / Pure White Linen Pink Coral / 출시년도 정보 없음	주어진 형식에 맞게 잘 출력하는 것을 확인 이미지 센터 크롭 적용, 목차별 구분과 구분선 표시, 향 표현등의 강조할 표현에 색표현과 볼드로 강조를 하되 어떤향인가요 같은 소재목과는 다른 색으로 표현	32.52s

TEST-ID	테스트 목적	상세 목적 및 예상 시나리오	테스트 환경	입력 데이터(값)	기대 출력	실제 출력	분석 및 개선점	응답 소요 (초)
LLM-W-005	향을 설명할때 전문적인 용어를 사용하지는 않는지	사용자 입력 "여름에 여자가 쓸 상큼한 향수 추천해줘" 입력 → 추천 설명 전반에 플로럴, 노트, 탑/미들/베이스 등의 전문적이 표현이 포함되지 않는지 확인	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-5.2	"여름에 여자가 쓸 상큼한 향수 추천해줘"	어떤 향인가요?: 처음엔 과일속에서 터지는 후추 한 방울처럼 경쾌하게 터져요. 곧 자스민의 꽃향기가 부드럽게 올라오면서, 마지막엔 부드러운 나무향 으로 포근하게 정리해줘요.	어떤 향인가요?: 시작은 귤껍질을 살짝 비틀어 툭 터프린 듯한 상큼함 이 먼저 와요. 시간이 지나면 하얀 꽃의 깨끗한 향 이 조용히 올라오고, 끝에는 이끼 낀 숲 바닥 같은 보송보송한 초록 기운 과 온근한 달콤함 이 피부에 얹게 남아요.	실제로 노트나 어코드 같은 전문적인 표현을 직접적으로 사용하지 않고 사용자가 이해하기 쉬운 표현으로 치환하여 표시하고 있는 것을 확인 - 머스크, 레몬등의 일상적으로 사용하게 되는 표현의 경우 사용이 허용되는데 머스크 등의 일부 표현의 경우 대중적이다의 기준이 모호하여 정의가 필요함	32.52s
LLM-W-006	추천의 이유를 사용자의 쿼리와 잘 이어서 제시하는지	사용자 입력 "20대 여성에게 선물할 향수를 추천해줘" 입력 → 재질문 → 사용자 입력"귀여운 이미지" 입력 → 추천의 이유를 설명할때 20대 여성에게 선물할 목적, 귀여운 이미지에 맞는 키워드가 선택되었는지 확인	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-5.2	"20대 여성에게 선물할 향수를 추천해줘", "귀여운 이미지"	추천 이유: 귀여운 20대 여성분에게는 상큼하고 달콤한 향이 잘 어울립니다. 그래서 새콤달콤한 과일향을 가진 향수를 추천해드려요.	추천 이유: "귀여운" 이미지를 가장 직관적으로 만들기 쉬운 타입이예요. 20대 여성에게 잘 어울리는 상큼달콤 과일 사탕 느낌 이라, 향수 초보도 실패 확률이 낮아요.	추천이유를 사용자의 쿼리와 잘 연결지어서 설명 - 이전 프롬프트에서는 20대의 여성과 같은 대상에 관한 직접적인 언급이 들어갔었는데 프롬프트 수정과 정에서 결과도 함께 바뀐 것으로 추정 이미지뿐만 아니라 해당 대상에 관한 분석과 이유를 제시하도록 프롬프트 강화 예정	3.23s + = 30.05
LLM-W-007	향의 변화를 탑/미들/베이스 순으로 시간의 흐름에 따라 설명하는지	사용자 입력 "여름에 여자가 쓸 상큼한 향수 추천해줘" 입력 → 향의 표현을 설명하는 부분에 ~~느낌으로 시작되어 시간이 좀 지나면 ~~느낌이 강해지고 ~~향이 은은하게 남습니다. 의 형태로 표현되는지 확인	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-5.2	"여름에 여자가 쓸 상큼한 향수 추천해줘"	어떤 향인가요?: 처음엔 과일속에서 터지는 후추 한 방울처럼 경쾌하게 터져요. 곧 자스민의 꽃향기가 부드럽게 올라오면서, 마지막엔 부드러운 나무향 으로 포근하게 정리해줘요.	어떤 향인가요?: 시작은 자몽처럼 산뜻한 과일 향 이 가볍게 튀고, 이어서 벚꽃-라일락·작약 같은 꽃향이 비누 거품처럼 뽀얗고 깔끔하게 퍼져요. 마지막엔 하얀 머스크 가 남아 막 사워한 사람 같은 청결감 이 오래가요.	실제로 탑/미들/베이스 노트가 각각 [Top]: Currant, Passion fruit, Pink grapefruit [Middle]: Cherry blossom, Lilac, Peony, Rose [Base]: Cedar, White musk 로 각각의 요소를 시간 순서대로 정리하여 제시 - 여러 요소가 섞여있는 경우 특정 향들로 줄이는 경향성이 있는 것으로 파악되어 제약 조건에 관한 논의 필요	26.82s
LLM-W-008	전혀 상관없는 이야기가 들어왔을 때 잘 대처하는지 (필요에 따라 별도의 노트로 구성)	사용자 입력 "오늘 날씨가 어때?" → 응답"아쉽지만 저는 날씨를 알 수 없습니다. 혹시 오늘은 따뜻한가요? 아니면 쌀	Docker Container (Python 3.11), PostgreSQL 17 (pgvector), gpt-5.2	"오늘 날씨 어때?"	"아쉽지만 저는 날씨를 알 수 없습니다. 혹시 오늘은 따뜻한가요? 아니면 쌀쌀한가요? 알려주시면 오늘	앗, 저는 서버 안에서만 살아서 오늘 바깥 날씨를 실시간으로 못 봤어요—미안해요. 🥲 지금 계산 곳은 맑	기본적인 내용들은 대답이 가능하지만 출력 형식이 앳, 저는 서버에만 살아서 ~~ 형태로 제시되는 느낌이	2.46s

TEST-ID	테스트 목적	상세 목적 및 예상 시나리오	테스트 환경	입력 데이터(값)	기대 출력	실제 출력	분석 및 개선점	응답 소요(초)
		썰한가요? 알려주시면 오늘에 어울리는 향수를 추천해드릴게요!"의 형식의 답변을 출력			에 어울리는 향수를 추천해드릴게요!"	아요, 흐려요, 비와요?	강해서 좀 더 자연스러운 대화가 가능하도록 프롬프트 수정 예정	

7. 결론 및 향후 개선 방향

• 결과 요약:

- **다단계 검색(Multi-stage Fallback) 기반 RAG 시스템 구축:** 단순 벡터 검색의 한계를 보완하기 위해 **[정확 일치(Exact) -> 오타 교정(Levenshtein) -> 의미 검색(Vector)]** 순으로 이어지는 안정적인 검색 파이프라인을 구축. 이를 통해 사용자의 다양한 질의 유형에 대응하는 서비스의 기술적 기초 마련.
- **모델 및 비용 효율성 최적화:** `gpt-4o-mini` 와 `text-embedding-3-small` 조합을 통해, 높은 수준의 감성 추론 성능을 유지하면서도 API 호출 비용을 최소화하여 실제 서비스 운영이 가능한 경제적 구조를 확보.
- **정량적 성능 검증 완료:** 엄격한 커스텀 지표 평가를 통해 **Fact Accuracy(0.77)** 및 **Emotional Grounding(0.76)** 수치를 도출됨. 이는 시스템이 단순한 정보 요약을 넘어, 성분 데이터에 기반한 고차원적 감성 비유를 성공적으로 수행하고 있음을 정량적으로 입증.

• 한계 및 개선점:

- **모델 외부 지식의 간섭 (Fact Accuracy 0.77):** 모델이 답변 생성 시 DB에 명시되지 않은 향수 상식을 섞어 설명하는 경향이 포착됨. 이는 사용자에게 풍부한 설명을 제공하는 장점이 있으나, 엄격한 팩트 체크 관점에서는 향후 프롬프트 제어를 통한 고도화 필요.
- **검색 엔진의 정밀도 한계:** 현재의 순차적 폴백(Fallback) 방식은 키워드 연관도와 의미적 유사도를 동시에 고려한 통합 랭킹 산출에 한계가 있어, 일부 감성 질의에서 검색 결과의 관련성이 변동되는 현상 발견.
- **데이터 노이즈:** 향수 리뷰 데이터 내에 포함된 주관적인 농담이나 비유적 표현들이 임베딩 품질을 저하시켜, 검색 정밀도를 낮추는 요인으로 파악됨.

• 향후 개발 계획 및 방향:

- 리랭킹(Reranking) 및 데이터 정제 고도화
 - **내용:** 검색된 후보군에 대해 **Cross-Encoder 기반의 리랭킹** 단계를 추가하고, LLM을 활용해 DB 내 리뷰 데이터의 노이즈를 제거하는 전처리 과정을 강화.
 - **목표:** 데이터의 순도를 높이고 추천의 최종 정밀도를 최상위 수준으로 끌어올림.
- 사용자 친화적 멀티모달(Multimodal) 확장.
 - **내용:** 텍스트 질의 외에도 사용자가 원하는 분위기의 이미지(사진, 색감 등)를 입력하면 이를 분석하여 어울리는 향수를 추천하는 기능을 검토.
 - **목표:** 언어로 표현하기 어려운 미묘한 '감성'을 시각적 데이터를 통해 더 직관적으로 반영.
- '향기 맵(Scent Map)' 시각화로 추천의 설득력 확보.
 - **내용:** 추천 결과가 도출된 기술적 근거를 사용자가 직관적으로 이해할 수 있도록 시각적 피드백 시스템을 구축. 추천된 향수의 주요 노트(Top, Middle, Base)와 사용자의 질문 키워드 간의 상관관계를 그래프나 맵 형태로 제공.
 - **목표:** 단순히 결과만 제시하는 '블랙박스'형 추천에서 벗어나, "왜 이 향수가 나에게 어울리는지"에 대한 객관적 근거를 시각화하여 서비스의 신뢰도와 사용자 만족도를 제고.