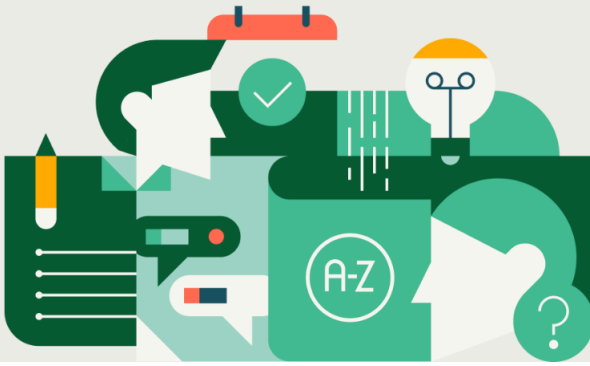
databricks

Databricks Exam 가이드

# Databricks Certified Generative AI Engineer Associate




시험 가이드 피드백 제공

## 정보 업데이트

| 업데이트 | 날짜 |
|---|---|
| 시험 출시 | 2026년 3월 18일 |
| 시험 가이드 업데이트 | 2026년 3월 18일 |

시험 날짜 2주 전에 당사 웹페이지의 시험 가이드를 확인하여 현재 버전의 시험을 준비하고 있는지 확인하십시오.

## 이 시험 가이드의 목적

이 시험 가이드의 목적은 시험에 대한 개요와 시험 범위를 제공하여 시험 준비 상태를 판단하는 데 도움을 드리는 것입니다. 이 문서는 시험에 변경 사항이 있을 때마다 업데이트되며(변경 사항이 적용되는 시기도 함께 표시됩니다), 이를 통해 확신을 가지고 시험을 준비하실 수 있습니다. 이 가이드는 위에 명시된 발효일 기준으로 현재 시행 중인 시험 버전을 다룹니다.

## 대상 독자 설명

Databricks Certified 생성형 AI Engineer Associate 자격증 시험은 Databricks를 사용하여 LLM 기반 솔루션을 설계하고 구현하는 개인의 역량을 평가합니다. 여기에는 복잡한 요구 사항을 관리 가능한 작업으로 분해하는 문제 분해 능력과 함께, 포괄적인 솔루션 개발을 위해 현재 생성형 AI 환경에서 적절한 모델, 도구 및 접근 방식을 선택하는 능력이 포함됩니다. 또한 시맨틱 유사도 검색을 위한 Vector Search, 모델 및 솔루션 배포를 위한 Model Serving, 솔루션 라이프사이클 관리를 위한 MLflow, 데이터 거버넌스를 위한 Unity Catalog 등 Databricks 고유의 도구도 평가합니다. 이 시험에 합격한 개인은 Databricks와 그 도구 세트를 최대한 활용하는 고성능 RAG 애플리케이션 및 LLM 체인을 구축하고 배포할 수 있을 것으로 기대됩니다.

## 시험 소개

| 항목 | 세부 정보 |
| --- | --- |
| 채점 문항 수 | 채점되는 객관식 및 다중 선택 45문항 |
| 채점 제외 문항 포함 가능성 | 최대 10개의 채점 제외 문항이 포함될 수 있음(아래 "채점 제외 콘텐츠" 참조) |
| 표시될 수 있는 총 문항 수 | 채점 문항 45개 및 채점 제외 문항 |
| 제한 시간 | 90분(채점 제외 문항을 고려한 추가 시간이 이미 반영됨) |
| 응시료 | USD 200, 현지 법률에 따라 부과되는 세금 별도 |
| 시험 응시 방식 | 온라인 감독 |
| 제공 언어 | 영어, 일본어, 브라질 포르투갈어(pt-BR), 한국어 |
| 시험 보조 도구 | 허용되지 않음 |
| 사전 요건 | 필수 요건 없음, 단 관련 교육 이수 적극 권장 |
| 권장 경험 | 명시된 시험 개요에 따라 Databricks Platform에서의 6개월 실무 경험이 적극 권장됩니다 |
| 유효 기간 | 2년 |

재인증: 인증된 상태를 유지하려면 **[2]**년마다 재인증이 필요합니다. 재인증을 받으려면 현재 시행 중인 시험에 응시해야 합니다.

채점되지 않는 콘텐츠: 시험에는 향후 활용을 위한 통계적 정보 수집을 목적으로 채점되지 않는 문항이 포함될 수 있습니다. 이러한 문항은 시험에서 별도로 표시되지 않으며 점수에 영향을 미치지 않고, 해당 문항을 고려하여 추가 시간이 제한 시간에 반영됩니다.

## 권장 준비 사항

### 학습

- 강사 주도: **Generative AI** 엔지니어링 **with Databricks**
- 자기 주도 학습 **(Databricks Academy**에서 이용 가능**):**
    - Agent Bricks로 RAG Agents 구축하기
    - Databricks에서 에이전틱 애플리케이션 구축하기
    - Agent Evaluation on Databricks
    - Databricks에서 에이전트 애플리케이션 배포 및 모니터링

## AI Prep 가이드 - 모든 Databricks 시험

Databricks AI Prep Guide 가이드는 이미 사용 중인 AI 챗봇을 Databricks 자격증 시험을 위한 맞춤형 튜터로 활용하는 방법을 안내합니다. 공식 시험 가이드를 활용하여 챗봇을 준비시키고, AI가 제공하는

오래된 답변을 파악하며, 자신의 취약점을 발견하고, 모든 Databricks 시험에서 요구되는 핸즈온 실습으로 넘어갑니다.

## 핸즈온 실습

무엇을 연습해야 할지 모르시겠습니까? AI Prep 가이드(위의 AI Prep 가이드 섹션 참조)에는 "Step 5 — Hands-on minimum" 아래에 프롬프트가 포함되어 있으며, 이 프롬프트를 통해 AI 챗봇이 이 시험의 목표에 맞게 매핑된 6~10개의 실습 작업으로 구성된 개인화된 체크리스트를 생성합니다. 각 작업은 Databricks .Free Edition에서 완료할 수 있을 만큼 충분히 작습니다

## Databricks 문서

시험을 준비하면서 이 시험에서 다루는 플랫폼 영역에 대한 공식 Databricks 문서를 학습하십시오. 공식 Databricks 문서는 가장 최신의 권위 있는 소스입니다. 이를 활용하여 각 주제를 얼마나 깊이 학습해야 하는지 판단하고, 오래된 자료가 아닌 최신의 더 이상 사용되지 않는 기능을 학습하고 있는지 확인하십시오. 공식 문서: Databricks 문서

## 시험 개요

### 섹션 1: 애플리케이션 설계 (14%)

- 특정 형식의 응답을 유도하는 프롬프트를 설계합니다
- 주어진 비즈니스 요구 사항을 달성하기 위한 모델 작업을 선택합니다
- 원하는 모델 입력 및 출력에 맞는 체인 구성 요소를 선택합니다.
- 비즈니스 사용 사례 목표를 AI 파이프라인의 원하는 입력 및 출력에 대한 설명으로 변환합니다.
- 다단계 추론을 위해 지식을 수집하거나 작업을 수행하는 도구를 정의하고 순서를 지정합니다.
- 문제 해결을 위해 Agent Bricks(Knowledge Assistant, Multiagent Supervisor, Information Extraction)를 사용하는 방법과 시기를 결정합니다.

### Section 2: Data Preparation (14%)

- 주어진 문서 구조와 모델 제약 조건에 맞는 청킹 전략을 적용합니다
- RAG 애플리케이션의 품질을 저하시키는 소스 문서의 불필요한 콘텐츠를 필터링합니다.
- 제공된 소스 데이터에서 문서 내용을 추출하고 형식을 지정하는 데 적합한 Python 패키지를 선택하십시오.
- 청크된 텍스트를 Unity Catalog의 Delta Lake 테이블에 쓰기 위한 운영 및 순서를 정의합니다.
- 특정 RAG 애플리케이션에 필요한 지식과 품질을 제공하는 소스 문서를 식별합니다.
- 도구와 메트릭을 사용하여 검색 성능을 평가합니다
- 고급 청킹 전략을 사용하여 검색 시스템을 설계합니다
- 정보 검색 프로세스에서 재순위화(re-ranking)의 역할을 설명하십시오.

## Section 3: Application Development (30%)

- 생성형 AI 애플리케이션에서 사용할 LangChain/유사 도구를 선택합니다.
- 품질 및 안전성과 같은 일반적인 문제를 파악하기 위해 응답을 정성적으로 평가합니다
- 모델 및 검색 평가를 기반으로 청킹 전략 선택
- 키 필드, 용어 및 의도를 기반으로 사용자 입력의 추가 컨텍스트를 프롬프트에 보강합니다.
- LLM의 응답을 기준선에서 원하는 출력으로 조정하는 프롬프트를 작성합니다
- 부정적인 결과를 방지하기 위해 LLM 가드레일을 구현합니다
- 개발할 애플리케이션의 속성을 기반으로 최적의 LLM을 선택합니다
- 소스 문서, 예상 쿼리 및 최적화 전략을 기반으로 임베딩 모델 컨텍스트 길이를 선택합니다.
- 모델 메타데이터/모델 카드를 기반으로 태스크에 적합한 모델을 모델 허브 또는 마켓플레이스에서 선택합니다.
- 실험에서 생성된 일반적인 메트릭을 기반으로 주어진 태스크에 가장 적합한 모델을 선택합니다
- 에이전트 시스템 개발을 위해 MLflow와 Agent Framework를 활용합니다
- Gen AI 애플리케이션 수명 주기의 평가 단계와 모니터링 단계를 비교합니다.
- 멀티 에이전트 시스템이 Genie Spaces 또는 conversational API를 활용하여 데이터를 검색할 수 있도록 지원합니다.

## 섹션 4: 애플리케이션 조립 및 배포 (22%)

- 전처리 및 후처리가 포함된 pyfunc 모델을 사용하여 체인을 코딩합니다
- Model Serving 엔드포인트에서 리소스에 대한 액세스를 제어합니다
- 요구 사항에 따라 간단한 체인을 코딩합니다
- RAG 애플리케이션을 만드는 데 필요한 기본 요소를 선택하십시오: 모델 flavor, 임베딩 모델, retriever, 종속성, 입력 예시, 모델 서명
- MLflow를 사용하여 Unity Catalog에 모델을 등록합니다.
- Vector Search 인덱스 생성 및 쿼리
- Foundation Model APIs를 활용하는 LLM 애플리케이션을 서빙하는 방법을 파악합니다.
- Mosaic AI Vector Search의 주요 개념과 구성 요소를 설명합니다.
- 배치 추론 워크로드를 식별하고 ai_query()를 적절하게 적용합니다.
- 임베딩 수, 업데이트 빈도, 지연 시간 및 비용 요구 사항을 기반으로 특정 솔루션에 맞게 Vector Search를 구성합니다.
- 중간 메모리 또는 구조화된 정보를 저장하고 검색하기 위한 영구 데이터 저장소를 구성합니다.
- Vector Search 인덱스 업데이트, 환경 간 프롬프트 승격, 에이전트의 개별 구성 요소 테스트 등 CI/CD 모범 사례를 적용합니다.
- 주어진 애플리케이션 요구 사항에 따라 관리형, 외부, 사용자 지정 MCP 서버를 통합합니다.
- 프롬프트 버전 제어를 적용하고 프롬프트 라이프사이클을 관리합니다.
- 에이전트 사용 시나리오에 적합한 대화형 사용자 인터페이스를 개발합니다(Apps, Slack, Teams 등).

## Section 5: Governance (8%)

- 성능 목표를 달성하기 위한 가드레일로 마스킹 기법을 사용합니다

- Gen AI 애플리케이션에 대한 악의적인 사용자 입력을 방지하기 위한 가드레일 기법을 선택합니다.
- 법적 위험을 방지하기 위해 데이터 소스에 대한 법적/라이선스 요건을 활용합니다.
- 생성형AI 애플리케이션에 데이터를 제공하는 데이터 소스에서 문제가 있는 텍스트를 완화하기 위한 대안을 권장합니다

### 섹션 6: 평가 및 모니터링 (12%)

- 정량적 평가 메트릭 집합을 기반으로 LLM 선택(크기 및 아키텍처)을 선정합니다.
- 특정 LLM 배포 시나리오에 대해 모니터링할 주요 메트릭을 선택합니다.
- MLflow 스코어링 및 트레이싱을 사용하여 에이전트 성능을 평가합니다
- 추론 로깅을 사용하여 배포된 RAG 애플리케이션의 성능을 평가합니다
- Databricks 기능을 사용하여 LLM 비용을 제어합니다
- 추론 테이블과 Agent Monitoring을 사용하여 라이브 LLM 엔드포인트를 추적합니다.
- 정답(ground truth)이 필요한 평가 판정자(judge)를 식별합니다
- Agent Framework를 통해 배포된 LLM 또는 에이전트를 추적하기 위해 AI Gateway(Inference Tables, Usage Tables, 속도 제한)를 사용합니다.
- 에이전트 및 LLMs 평가를 위해 Databricks 사용자 지정 Scorer를 사용합니다
- SME 피드백을 통합하여 에이전트 성능을 향상시킵니다

## 샘플 문제

샘플 문제의 목적은 시험 가이드에 명시된 목표를 보여주고 각 목표에 부합하는 샘플 문제를 제공하는 것입니다. 시험 가이드에는 시험에서 다룰 수 있는 목표가 나열되어 있으며, 가장 좋은 준비 방법은 위의 시험 날짜에 해당하는 전체 시험 개요를 검토하는 것입니다.

### 질문 1

목표**:** 주어진 문서 구조와 모델 제약 조건에 맞는 청킹 전략을 적용합니다

생성형 AI 엔지니어가 최대 1억 개를 지원하는 벡터 데이터베이스에 1억 5천만 개의 임베딩을 로드하려고 합니다. 레코드 수를 줄이기 위해 취할 수 있는 두 가지 조치는 무엇입니까?

A. 문서 청크 크기를 늘립니다
B. 청크 간의 겹침을 줄입니다
C. 문서 청크 크기를 줄입니다
D. 청크 간의 겹침을 늘립니다
E. 더 작은 임베딩 모델을 사용합니다

### 질문 2

목표**:** 주어진 ***RAG*** 애플리케이션에 필요한 지식과 품질을 제공하는 소스 문서를 식별합니다**.**

생성형 AI 엔지니어가 자동차 부품 판매를 지원하기 위해 개발 중인 고객 대상 생성형AI 애플리케이션의 응답을 평가하고 있습니다. 이 애플리케이션은 질문에 답변하기 위해 고객이 account_id와 transaction_id를 명시적으로 입력하도록 요구합니다. 출시 후 주문 및 청구 세부 정보에 대해서는 잘 처리했지만 배송 및 예상 도착일 관련 질문에는 정확하게 답변하지 못했다는 피드백이 있었습니다. 이 애플리케이션의 해당 질문에 대한 답변 능력을 향상시키기 위한 접근 방식은 무엇입니까?

A. 모든 자동차 부품에 대한 회사 배송 정책 및 결제 조건을 포함하는 벡터 스토어를 생성합니다
B. transaction_id를 기본 키로 사용하고 송장 데이터와 예상 배송 날짜로 채워진 특성 저장소 테이블을 생성합니다.
C. 예상 도착 날짜에 대한 예시 데이터를 튜닝 데이터셋으로 제공한 다음, 모델이 최신 배송 정보를 갖출 수 있도록 주기적으로 파인 튜닝합니다.
D. 주문이 접수된 시점을 입력으로 받도록 채팅 프롬프트를 수정하고, 어떤 배송 방법도 14일을 초과하지 않으므로 모델이 해당 날짜에 14일을 추가하도록 지시합니다.

## 질문 3

목표**:** 제공된 소스 데이터에서 문서 내용을 추출하고 형식을 지정하기 위한 적절한 ***Python*** 패키지를 선택합니다**.**

생성형 AI 엔지니어가 .jpeg 또는 .png와 같은 형식의 이미지 파일로 스캔되어 저장된 소스 문서에서 검색된 컨텍스트를 기반으로 하는 RAG 애플리케이션을 개발하고 있습니다. 최소한의 코드로 솔루션을 개발하려고 합니다. 소스 문서에서 텍스트를 추출하려면 어떤 Python 패키지를 사용해야 합니까?

A. beautifulsoup
B. scrapy
C. pytesseract
D. pyquery

## 질문 4

목표**:** 소스 문서**,** 예상 쿼리 및 최적화 전략을 기반으로 임베딩 모델 컨텍스트 길이 선택

생성형 AI 엔지니어가 LLM 기반 애플리케이션을 개발하고 있습니다. 검색기에 사용할 문서는 각각 최대 512 토큰으로 청크 분할되어 있습니다. 엔지니어는 이 애플리케이션에서 품질보다 비용과 지연 시간이 더 중요하다는 것을 알고 있습니다. 선택할 수 있는 컨텍스트 길이 수준이 여러 가지 있을 때, 요구 사항을 충족하는 것은 무엇입니까?

A. context length 512: 가장 작은 모델은 0.13GB이며 embedding dimension은 384입니다.
B. context length 514: 가장 작은 모델은 0.44GB이며 임베딩 차원은 768입니다
C. context length 2048: 가장 작은 모델은 11GB이며 임베딩 차원은 2560입니다
D. context length 32768: 가장 작은 모델은 14GB이며 임베딩 차원은 4096입니다

## 질문 5

목표**:** 개발할 애플리케이션의 속성을 기반으로 최적의 ***LLM*** 선택하기

생성형 AI 엔지니어가 한 단락 정도 길이의 메모 필드를 메모의 의도를 나타내면서 애플리케이션 프런트엔드에 맞는 단 한 문장의 요점으로 업데이트할 수 있는 애플리케이션을 구축하려고 합니다. 이 애플리케이션에 적합한 LLMs를 평가할 때 어떤 자연어 처리 태스크 카테고리를 사용해야 합니까?

A. text2text Generation
B. Sentencizer
C. 텍스트 분류
D. 요약

## Question 6

목표**:** 임베딩 수**,** 업데이트 빈도**,** 지연 시간 및 비용 요구 사항을 기반으로 특정 솔루션에 맞게 ***Vector Search***를 구성합니다**.**

온라인 소매업체에서 근무하는 생성형 AI 엔지니어가 Vector Search와 메타데이터 필터링을 활용하여 검색 기능을 개선하려고 합니다. 검색은 초당 최대 80건까지 발생할 수 있으며, 지연 시간이 가장 중요한 지표입니다. 정확도를 높이고 지연 시간에 영향을 주지 않는다면 초기 개발 비용은 감수할 의향이 있습니다. 재고는 전국적으로 1억 개의 항목으로 구성되어 있습니다. 엔지니어는 이를 어떻게 설정해야 합니까?

A. GTE Large 임베딩 모델을 활용하고, 하이브리드 검색과 재순위화를 활성화한 표준 Vector Search를 사용합니다.
B. GTE Large 임베딩 모델을 활용하고, 하이브리드 검색 및 재순위화를 활성화한 스토리지 최적화 Vector Search를 사용합니다.
C. 사용자 정의 임베딩 모델을 파인 튜닝하고, 표준 Vector Search를 사용하며, 하이브리드 검색과 리랭킹은 비활성화 상태로 유지합니다.
D. 사용자 정의 임베딩 모델을 파인 튜닝하고, 스토리지 최적화 Vector Search를 사용하며, 하이브리드 검색과 리랭킹은 비활성화 상태로 유지합니다.

## 질문 7

목표***: Vector Search*** 인덱스 업데이트**,** 환경 간 프롬프트 승격**,** 에이전트의 개별 구성 요소 테스트 등 ***CI/CD*** 모범 사례를 적용합니다**.**

생성형 AI 엔지니어가 개발, 스테이징, 프로덕션 환경 전반에 걸쳐 에이전트의 프롬프트 템플릿을 관리해야 합니다. 팀은 게이트 릴리스 프로세스를 요구합니다. 즉, 프롬프트는 개발 환경에서 업데이트되고, 자동화된 테스트를 통해 스테이징에서 검증된 후, 승인을 거친 경우에만 프로덕션 환경으로 승격됩니다. 이 솔루션은 버전 이력을 보존하고 필요한 경우 이전 프롬프트 버전으로 롤백할 수 있어야 합니다. 이 승격 워크플로를 지원하는 접근 방식은 무엇입니까?

A. 테스트가 통과된 후 스테이징 브랜치를 프로덕션 브랜치에 병합하여 프롬프트 템플릿을 애플리케이션 리포지토리에 저장하고 승격합니다.

B. 프롬프트를 MLflow 버전으로 추적하고, 통과 후 별칭을 사용하여 프롬프트를 승격합니다.
C. CI 러너의 JSON 파일에 프롬프트를 저장하고 각 실행 시마다 프로덕션 프롬프트를 덮어씁니다.
D. 일관성을 보장하기 위해 프롬프트를 Delta 테이블에 저장하고 배포할 때마다 프로덕션 테이블을 덮어씁니다.

## 질문 8

목표**:** 에이전트 사용 시나리오에 적합한 대화형 사용자 인터페이스를 개발합니다 ***(Apps, Slack, Teams*** 등***).***

생성형 AI 엔지니어가 고객 지원 담당자가 내부 PDF를 기반으로 질문하고 답변을 받을 수 있는 Databricks App을 구축하고 있습니다. 요구 사항은 다음과 같습니다. 사용자는 회사 계정(corporate identity)으로 인증해야 하고, 앱은 브라우저에 장기 토큰을 노출하지 않고 Mosaic AI Agent 엔드포인트를 호출해야 하며, 답변에 대한 접근은 각 사용자의 권한을 준수해야 합니다. 이러한 요구 사항을 충족하는 접근 방식은 무엇입니까?

A. Databricks App 백엔드를 사용하여 앱의 자격 증명으로 Agent 엔드포인트를 호출하고, 앱의 인증된 컨텍스트를 통해 사용자 ID/권한을 적용합니다.
B. 앱의 JavaScript에 Databricks 개인 액세스 토큰(PAT)을 저장하고 브라우저에서 직접 Agent 엔드포인트를 호출합니다.
C. Agent 엔드포인트를 공개적으로 게시하고 앱 프론트엔드에 포함된 API 키로 보호합니다.
D. Agent가 신원 확인 없이 읽을 수 있도록 PDF를 공개 버킷으로 내보냅니다.

## Question 9

목표**:** 주어진 애플리케이션 요구 사항에 따라 관리형**,** 외부**,** 커스텀 ***MCP*** 서버를 통합합니다

생성형 AI 엔지니어가 인터넷 데이터 소스에서 사실적인 정보에 접근하고 외부 API를 사용하여 웹 검색을 수행해야 하는 리서치 어시스턴트 에이전트를 구축하고 있습니다. Databricks는 이 인터넷 데이터 소스에 대한 관리형 MCP 서버를 제공하며, 키가 필요한 외부 API를 위한 외부 MCP 서버도 사용 가능합니다. 이 애플리케이션은 프로덕션 환경에서 두 데이터 소스에 대한 안정적인 접근을 보장하면서 유지 관리 오버헤드를 최소화해야 합니다. 엔지니어가 이러한 데이터 소스를 에이전트에 통합하기 위해 취해야 할 두 가지 조치는 무엇입니까?

A. 인터넷 리소스와 외부 API를 모두 래핑하여 에이전트가 호출할 수 있는 단일 통합 인터페이스로 제공하는 커스텀 MCP 서버를 구축합니다.
B. 관리형 웹 브라우저 MCP 서버를 사용하여 프로그래밍 방식으로 인터넷 리소스를 탐색하고 정보를 검색합니다.
C. 에이전트의 오프라인 액세스를 위해 인터넷 리소스의 콘텐츠와 검색 결과를 캐시하도록 Unity Catalog 외부 테이블을 구성합니다.
D. 에이전트의 MCP 서버 구성을 통해 서버 유형을 "managed"로 지정하고 인터넷 리소스의 서버 식별자를 제공하여 관리형 MCP 서버를 구성합니다.
E. 외부 MCP 서버의 연결 세부 정보를 제공하고, API 키를 Databricks Secrets에 저장한 후, MCP 서버 구성에서 이를 참조하여 외부 MCP 서버를 배포합니다.

## 질문 10

목표***: SME*** 피드백을 통합하여 에이전트 성능 개선

생성형 AI 엔지니어가 운영 팀에서 내부적으로 사용하는 고객 지원 RAG 어시스턴트를 평가하는 업무를 담당하고 있습니다. 4명의 도메인 전문가가 매주 MLflow에서 샘플링된 답변을 사실 정확성, 완전성, 유용성 등의 기준으로 검토합니다. 여러 차례 반복 후, 엔지니어는 동일한 응답에 대해 전문가 평점이 크게 다르다는 것을 발견하였으며, 이로 인해 평가 데이터가 시간 경과에 따른 모델 개선 추적에 신뢰할 수 없게 되었습니다. 엔지니어는 평가자 간 불일치를 줄이면서 반복적인 품질 개선을 지원하는 신뢰할 수 있는 평가 프로세스가 필요합니다. 엔지니어는 어떻게 해야 합니까?

A. LLM-as-a-judge를 사용하여 과거 및 미래의 응답을 재채점하고, 전문가 간의 의견 불일치를 조정하는 대신 모델이 생성한 평가를 기준 정답으로 취급합니다.
B. 명확한 루브릭을 정의하고, 해당 기준에 따라 SME를 보정한 후, 일관된 agent evaluation을 위해 정렬된 판단 결과를 mlflow.genai.evaluate()에 활용합니다.
C. 각 응답에 대해 모든 도메인 전문가의 점수를 평균 내고, 혼합된 점수를 모델 튜닝의 최종 벤치마크로 직접 사용합니다.
D. 모든 전문가가 이미 동의한 응답만으로 벤치마크를 구축하고, 일관성을 높이기 위해 평가 세트에서 이견이 있는 사례를 제외합니다.

# 정답 키 및 해설

| # | 정답 | 설명 |
|---|---|---|
| 1 | A, B | 청크를 더 크게 하고 중첩을 줄이면 둘 다 전체 청크 수가 줄어들어 임베딩 수가 1억(100M) 한도 아래로 낮아집니다. 청크를 더 작게 하거나 중첩을 늘리면 수가 증가합니다. |
| 2 | B | 예상 배송일은 구조화된 거래별 사실 정보로, 정책을 담은 벡터 스토어나 파인튜닝이 아니라 transaction_id를 키로 하는 피처 스토어 테이블에서 제공하는 것이 가장 좋습니다. 고정된 14일을 더하는 방식은 부정확합니다. |
| 3 | C | 원본이 스캔된 이미지(.jpeg/.png)이므로 OCR이 필요하며, pytesseract는 최소한의 코드로 이미지 텍스트를 추출합니다. beautifulsoup, scrapy, pyquery는 OCR이 아니라 HTML/웹 스크래핑 도구입니다. |
| 4 | A | 청크가 512 토큰으로 제한되고 품질보다 비용/지연 시간이 더 중요하므로, 512 컨텍스트, 0.13GB, 384차원 모델이 가장 효율적입니다. 더 큰 컨텍스트/차원 모델은 여기서 이점 없이 비용과 지연 시간만 늘립니다. |
| 5 | D | 문단을 한 문장 요지로 압축하는 것은 요약(summarization) 작업이므로, 후보 LLM은 Summarization NLP 작업 범주에서 평가해야 합니다. |
| 6 | C | 지연 시간이 핵심 지표이므로 지연을 늘리는 하이브리드 검색과 리랭킹은 끕니다. 팀이 초기 비용을 감수하므로 파인튜닝된 맞춤형 임베딩 모델로 정확도를 높이고, 표준 벡터 검색으로 처리량 요구를 충족합니다. |
| 7 | B | 프롬프트를 MLflow 버전으로 추적하고 별칭(alias)으로 승격하면 완전한 버전 이력과 롤백을 갖춘 |

| | | |
|---|---|---|
| | | 통제된 워크플로를 얻을 수 있습니다. 브랜치 병합, JSON 덮어쓰기, Delta 덮어쓰기는 적절한 버전 관리를 잃게 됩니다. |
| 8 | A | Databricks App 백엔드는 토큰을 서버 측에 보관하고, 회사 아이덴티티로 사용자를 인증하며, 사용자별 권한을 적용하여 모든 요구 사항을 충족합니다. PAT/API 키를 브라우저에 포함하거나 데이터를 공개적으로 노출하는 것은 제약 조건을 위반합니다. |
| 9 | D, E | 인터넷 소스에 대해서는 에이전트의 MCP 구성을 통해 관리형 MCP 서버를 설정하고(D), 외부 MCP 서버는 API 키를 Databricks Secrets에 저장하여 배포합니다(E). 이렇게 하면 유지 관리를 최소화하면서 안정적인 운영 환경 액세스를 보장합니다. 맞춤형 래퍼나 외부 테이블 캐싱은 오버헤드를 추가합니다. |
| 10 | B | 평가자 간 불일치는 명확한 평가 기준(rubric)을 정의하고 SME(주제 전문가)를 보정한 뒤, 정렬된 판정을 mlflow.genai.evaluate()에 사용하여 해결합니다. 잡음이 많은 점수를 평균 내거나, 논쟁이 있는 사례를 제외하거나, LLM 판정자를 무조건 신뢰하는 것은 근본 원인을 해결하지 못합니다. |