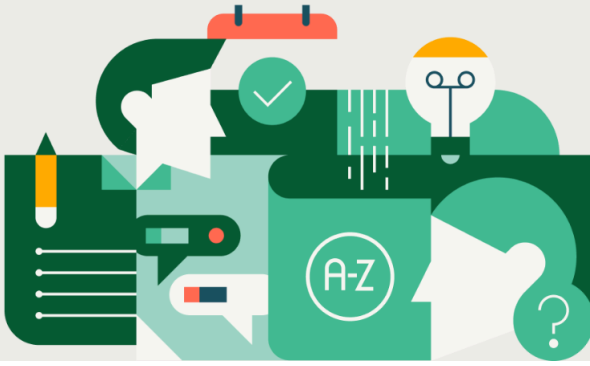
databricks

Databricks 시험 가이드

Databricks 인증
생성형 AI 엔지니어 어소시에이트



시험 가이드 피드백 제공

이 시험 가이드의 목적

이 시험 가이드는 시험 개요와 시험에서 다루는 내용을 제공하여 귀하의 시험 준비 여부를 판단하는 데 도움이 됩니다. 이 문서는 시험에 변경 사항이 생길 때마다, 그리고 그 변경 사항이 시험에 적용되는 시기에 맞춰 업데이트되어, 준비할 수 있도록 도와줍니다. 이 버전은 **2026**년 **3**월 **18**일 기준 현재 공개된 버전을 다룹니다**.** 시험을 보기 **2**주 전에 최신 버전을 꼭 확인해 주세요**.**

대상자 설명

Databricks 인증 생성형 AI 엔지니어 준회원 자격증 시험은 Databricks를 사용하여 LLM 지원 솔루션을 설계하고 구현하는 개인의 능력을 평가합니다. 여기에는 복잡한 요구사항을 관리 가능한 태스크로 분해하는 문제 분해와, 현재 생성형 AI 환경에서 적절한 모델, 도구, 접근법을 선택하여 포괄적인 솔루션을 개발하는 것이 포함됩니다. 또한 의미적 유사성 검색을 위한 Vector Search, 모델 및 솔루션 배포를 위한 Model Serving, 솔루션 수명주기 관리를 위한 MLflow, 데이터 거버넌스를 위한 Unity Catalog와 같은 Databricks 전용 도구도 평가합니다. 이 시험에 합격한 개인은 Databricks와 그 툴셋을 최대한 활용하는 성능 좋은 RAG 애플리케이션과 LLM 체인을 구축하고 배포할 것으로 기대됩니다.

시험 소개

- 채점된 문제의 수: 객관식 또는 선택 문항 45개*
  시간 제한: 90분
- 등록비: $200
- 전달 방법: 온라인 감독 시험
- 필수 조건: 필수는 아닙니다; 관련 강의 참석과 6개월간의 실무 경험을 강력히 권장합니다.
- 유효 기간: 2년.
- 재인증: 인증 상태를 유지하기 위해 2년마다 재인증이 필요합니다. 재인증을 받으려면 현재 진행 중인 전체 시험을 봐야 합니다. 시험 웹페이지의 "시험 준비하기" 섹션을 확인하여 다시 시험을 준비하세요.
- 점수 미지급 질문: 시험에는 미래에 사용할 통계 정보를 수집하기 위해 점수가 매기지 않은 문제들이 포함될 수 있습니다. 이 질문들은 양식에 명시되어 있지 않으며, 점수에 영향을 주지 않습니다. 이 문제들을 고려하기 위해 추가 시간이 시험에 반영됩니다.

권장 준비 방법

- 현재 Databricks Academy ILT에서 제공되는 모든 과정은 생성형 AI 학습자 역할과 관련되어 있으며, 특히 Databricks를 활용한 생성형 AI 엔지니어링 과정이 포함됩니다.
- 자율 학습 (Databricks Academy에서 제공) Databricks를 이용한 생성형 AI 엔지니어링, 다음 강좌들:
    - Databricks에서 검색 에이전트 구축
    - Databricks에서 단일 에이전트 애플리케이션 구축
    - 생성형 AI 애플리케이션 평가 및 거버넌스
    - 생성형 AI 애플리케이션 배포 및 모니터링
- 현재 LLM과 그 기능에 대한 지식
- 프롬프트 엔지니어링, 프롬프트 생성 및 평가에 대한 지식
- LangChain, Hugging Face Transformers 등 최신 온라인 도구와 서비스에 대한 지식
- Python과 RAG 애플리케이션, 에이전트, LLM 체인 개발을 지원하는 라이브러리에 대한 실무 지식
- 데이터 준비, 모델 체이닝 등을 위한 최신 API에 대한 실무 지식
- 관련 Databricks 문서 자료

시험 개요

섹션 **1:** 애플리케이션 설계

- 구체적으로 형식화된 답변을 이끌어내는 프롬프트를 설계하세요
- 주어진 비즈니스 요구사항을 달성하기 위해 모델 작업을 선택하세요
- 원하는 모델 입력과 출력에 맞는 연쇄 구성 요소를 선택하세요
- 비즈니스 활용 사례 목표를 AI 파이프라인에 필요한 입력과 출력의 설명으로 변환하세요
- 지식을 수집하거나 다단계 추론을 위해 행동을 수행하는 도구를 정의하고 순서를 정하세요 문제 해결을 위해 Agent Bricks(지식 보조자, 다중 에이전트 감독자, 정보 추출)를 언제, 어떻게 사용할지 결정하세요.

섹션 **2:** 데이터 준비

- 주어진 문서 구조와 모델 제약 조건에 대해 chunking 전략을 적용하세요
- RAG 애플리케이션의 품질을 저하시키는 소스 문서의 불필요한 콘텐츠를 필터링하세요. 제공된 소스 데이터와 형식에서 문서 내용을 추출하기 위해 적절한 Python 패키지를 선택하세요
- Unity Catalog에서 Delta Lake 테이블에 주어진 청크 텍스트를 쓰기 위한 연산과 순서를 정의하세요
- 주어진 RAG 애플리케이션에 필요한 지식과 품질을 제공하는 소스 문서를 식별하세요
- 도구와 메트릭을 활용해 검색 성능을 평가하세요
- 고급 청킹 전략을 사용한 검색 시스템 설계
- 정보 검색 과정에서 재순위 부여의 역할을 설명하세요

섹션 **3:** 애플리케이션 개발

- 생성형 AI 애플리케이션에 사용할 Langchain이나 유사한 도구를 선택하세요.
- 응답을 정성적으로 평가하여 품질 및 안전성과 같은 공통 문제를 파악합니다.
- 모델 및 검색 평가를 기반으로 청킹 전략을 선택합니다.
- 주요 필드, 용어, 의도를 바탕으로 사용자의 입력에서 추가 맥락을 포함하여 프롬프트를 보완하세요
- LLM의 응답을 기준선에서 원하는 출력으로 조정하는 프롬프트를 만드세요.
- 부정적인 결과를 방지하기 위한 LLM 가드레일을 구현하세요.
- 개발할 애플리케이션의 속성을 기준으로 최적의 LLM을 선택합니다.
- 원본 문서, 기대 쿼리, 최적화 전략을 기반으로 임베딩 모델 컨텍스트 길이를 선택합니다.
- 모델 메타데이터/모델 카드를 기반으로 태스크를 위해 모델 허브나 마켓플레이스에서 모델을 선택하세요
- 실험에서 생성된 일반적인 지표를 바탕으로 주어진 작업에 최적의 모델을 선택하세요
- 에이전트 시스템 개발을 위해 MLflow와 에이전트 프레임워크를 활용하세요.
- 생성형 AI 애플리케이션 수명주기의 평가 및 모니터링 단계를 비교하세요
- 멀티에이전트 시스템이 Genie Spaces 또는 대화형 API를 활용하여 데이터를 검색할 수 있도록 하세요

섹션 **4:** 애플리케이션 구성 및 배포

- pyfunc 모델을 사용하여 전처리와 후처리를 포함한 체인을 코딩하세요
- model serving 엔드포인트에서 리소스 접근을 제어합니다
- 요구사항에 따라 간단한 체인을 코딩합니다
- RAG 애플리케이션을 만들기 위해 필요한 기본 요소를 선택하세요: model flavor, 임베딩 모델, 리트리버, 의존성, 입력 예제, 모델 서명
- MLflow를 사용해 Unity Catalog에 모델을 등록하세요
- 벡터 검색 인덱스를 생성하고 쿼리하세요
- Foundation Model APIs를 활용하는 LLM 애플리케이션을 어떻게 제공할지 식별하세요
- Mosaic AI Vector Search의 핵심 개념과 구성 요소를 설명하세요
- 배치 추론 워크로드를 식별하고 `ai_query()`를 적절하게 적용합니다
- 임베딩 수, 업데이트 빈도, 지연 시간, 비용 요구사항을 기준으로 특정 솔루션에 대해 벡터 검색을 구성하세요.
- 중간 메모리나 구조화된 정보를 저장하고 불러오기 위해 영구 데이터스토어를 구성하세요.
- CI/CD 모범 사례를 적용하세요. 예를 들어, 벡터 검색 인덱스 업데이트, 환경 간 프롬프트 프로모트, 에이전트의 개별 구성 요소 테스트 등이 있습니다.
- 주어진 애플리케이션 요구사항에 따라 관리형, 외부형, 맞춤형 MCP 서버를 통합합니다
- 프롬프트 버전 관리를 적용하고, 프롬프트 수명주기를 관리하세요.
- 에이전트 사용 시나리오(앱, 슬랙, 팀즈 등)에 적합한 대화형 사용자 인터페이스를 개발하세요.

섹션 **5:** 거버넌스

- 마스킹 기법을 가드레일로 활용하여 성능 목표를 달성하기
- 생성형 AI 애플리케이션에 악의적인 사용자 입력을 방지하기 위한 가드레일 기법 선택하기

- 데이터 소스의 법적/라이선스 요건을 활용하여 법적 위험을 방지하기
- 생성형 AI 애플리케이션에 데이터를 제공하는 데이터 소스에서 문제성 텍스트 완화 대안 추천

섹션 **6:** 평가 및 모니터링

- 정량적 평가 메트릭 세트를 바탕으로 LLM(크기 및 아키텍처)을 선택하세요
- 특정 LLM 배포 시나리오에 대해 모니터링할 키 메트릭을 선택하세요
- MLflow 스코어링 및 트레이싱을 사용하여 에이전트 성능을 평가합니다
- 추론 로깅을 활용하여 배포된 RAG 애플리케이션 성능을 평가할 수 있습니다
- Databricks 기능을 활용하여 LLM 비용을 제어할 수 있습니다
- 추론 테이블과 에이전트 모니터링을 사용하여 실시간 LLM 엔드포인트를 추적합니다.
- 정답 데이터가 필요한 평가자를 식별합니다.
- AI 게이트웨이(추론 테이블, 사용 테이블, 속도 제한)를 사용하여 Agent Framework를 통해 배포된 LLM이나 에이전트를 추적하세요.
- Databricks 커스텀 스코어러를 사용하여 에이전트와 LLM을 평가하세요
- SME 피드백을 반영하여 에이전트 성능을 향상하세요

샘플 질문

이 문제들은 실제 문제 문항과 유사하며, 시험에서 어떻게 문제가 출제되는지 대략적으로 알 수 있게 해줍니다. 시험 가이드에 명시된 시험 목표와 해당 목표에 부합하는 샘플 문제를 제공합니다. 시험 가이드에는 시험에서 다룰 수 있는 모든 목표가 나열되어 있습니다. 자격증 시험을 준비하는 가장 좋은 방법은 시험 가이드의 시험 개요를 검토하는 것입니다.

질문 **1**

목적: 주어진 문서 구조 및 모델 제약 조건에 맞춰 청킹 전략을 적용합니다.
생성형 AI 엔지니어가 최대 1억 개의 임베딩을 처리할 수 있는 벡터 데이터베이스에 1억 5천만 개의 임베딩을 로드하고 있습니다.

레코드 수(데이터 개수)를 줄이기 위해 취할 수 있는 두 가지 조치는 무엇입니까?

A. 문서 청크 크기를 늘리기
B. 청크 간 겹침을 줄이기
C. 문서 청크 크기를 줄이기
D. 청크 간 겹침을 늘리기
E. 더 작은 임베딩 모델 사용

질문 **2**

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

생성형 AI 엔지니어가 자동차 부품 판매를 돕기 위해 개발 중인 고객 대상 생성형 AI 애플리케이션의 반응을 평가하고 있습니다. 이 애플리케이션은 고객이 질문에 답변하기 위해 `account_id`와 `transaction_id`을 명시적으로 입력해야 합니다. 초기 출시 후 고객 피드백은 애플리케이션이 주문 및 청구 세부 사항에 잘 응답했지만, 배송 및 예상 도착일 질문에는 정확히 답변하지 못했다는 것이었습니다.

다음 중 어떤 것이 애플리케이션의 이러한 질문에 대한 답변 능력을 향상시킬 수 있을까요?

A. 모든 자동차 부품에 대한 회사의 배송 정책 및 결제 조건을 포함하는 벡터 스토어를 생성합니다.
B. `transaction_id`를 기본 키(Primary Key)로 하고, 송장(Invoice) 데이터와 예상 배송일 정보가 채워진 피처 스토어(Feature Store) 테이블을 생성합니다.
C. 예상 도착 날짜에 대한 예시 데이터를 튜닝 데이터셋으로 제공하고, 모델이 최신 배송 정보를 가질 수 있도록 주기적으로 파인튜닝(Fine-tuning)을 수행합니다.
D. 주문이 접수된 날짜를 입력받도록 채팅 프롬프트를 수정하고, 어떤 배송 방법도 14일을 초과하지 않으므로 주문일로부터 14일을 더해 답변하도록 모델에 지시합니다.

질문 **3**

목적: 제공된 소스 데이터와 형식에서 문서 내용을 추출하기 위해 적절한 *Python* 패키지를 선택하세요.

생성형 AI 엔지니어가 .jpeg 또는 .png와 같은 형식의 이미지 파일로 스캔되어 저장된 소스 문서에서 컨텍스트를 검색하여 활용하는 RAG 애플리케이션을 개발하고 있습니다. 이 엔지니어는 가장 적은 코드 라인으로 솔루션을 개발하고자 합니다.

소스 문서에서 텍스트를 추출하려면 어떤 Python 패키지를 사용해야 할까요?

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

질문 **4**
목적: 소스 문서, 예상 쿼리, 최적화 전략을 바탕으로 임베딩 모델 컨텍스트 길이를 선택하세요

생성형 AI 엔지니어가 LLM 기반 애플리케이션을 만들고 있습니다. 리트리버 문서는 최대 512개의 토큰으로 분할되어 있습니다. 생성형 AI 엔지니어는 이 애플리케이션에서 품질보다 비용과 지연 시간이 더 중요하다는 것을 잘 알고 있습니다. 여러 가지 맥락 길이 레벨 중에서 선택할 수 있습니다.

어떤 것이 그들의 필요를 충족시킬까요?

A. 컨텍스트 길이 512: 가장 작은 모델은 0.13GB이며, 임베딩 차원은 384
B. 컨텍스트 길이는 514: 가장 작은 모델은 0.44GB이며 임베딩 차원은 768
C. 컨텍스트 길이 2048: 가장 작은 모델은 11GB이고 임베딩 차원은 2560
D. 컨텍스트 길이 32768: 가장 작은 모델은 14GB이며 임베딩 차원은 4096

질문 **5**

목적: 개발할 애플리케이션의 속성을 기준으로 최적의 *LLM*을 선택하세요

생성형 AI 엔지니어가 약 한 단락 분량의 메모 필드를, 애플리케이션 프런트엔드에 적합하도록 메모의 의도를 나타내는 단 한 문장의 요약(gist)으로 업데이트할 수 있는 애플리케이션을 구축하고자 합니다.

이 애플리케이션에서 어떤 자연어 처리 태스크 범주를 평가해야 합니까?

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

질문 **6**

목적: 임베딩 개수, 업데이트 빈도, 지연 시간 및 비용 요구 사항을 기반으로 특정 솔루션에 맞게 벡터 검색을 구성합니다.

온라인 소매업체에서 근무하는 생성형 AI 엔지니어가 벡터 검색과 메타데이터 필터링을 활용하여 검색 기능을 개선하려고 합니다. 초당 최대 80건의 검색이 발생하며, 지연 시간이 가장 중요한 요소입니다. 정확도를 높이면서 지연 시간을 줄일 수 있다면 초기 개발 비용은 감수할 수 있습니다. 해당 업체는 전국적으로 1억 개의 상품을 보유하고 있습니다.

엔지니어는 어떻게 설정해야 할까요?

A. GTE Large 임베딩 모델을 활용하고, 하이브리드 검색 및 재순위 지정을 활성화한 표준 벡터 검색을 사용합니다.
B. GTE Large 임베딩 모델을 활용하고, 하이브리드 검색 및 재순위 지정을 활성화한 저장 공간 최적화 벡터 검색을 사용합니다.
C. 사용자 지정 임베딩 모델을 미세 조정하고, 표준 벡터 검색을 사용하며, 하이브리드 검색 및 재순위 지정은 비활성화합니다.
D. 사용자 지정 임베딩 모델을 미세 조정하고, 저장 공간 최적화 벡터 검색을 사용하며, 하이브리드 검색 및 재순위 지정은 비활성화합니다.

질문 **7**

목적: 벡터 검색 인덱스 업데이트, 환경 간 프롬프트 배포, 에이전트 개별 구성 요소 테스트 등 *CI/CD* 모범 사례를 적용합니다.

생성형 AI 엔지니어는 개발, 스테이징, 프로덕션 환경 전반에 걸쳐 에이전트 프롬프트 템플릿을 관리해야 합니다. 팀은 단계별 릴리스 프로세스를 요구합니다. 프롬프트는 개발 환경에서 업데이트되고, 스테이징 환경에서 자동화된 테스트를 통해 검증된 후, 승인 후에만 프로덕션 환경으로 배포됩니다. 솔루션은 버전 기록을 유지하고 필요한 경우 이전 프롬프트 버전으로 롤백할 수 있어야 합니다.

어떤 접근 방식이 이러한 배포 워크플로를 지원합니까?

A. 프롬프트 템플릿을 애플리케이션 저장소에 저장하고, 테스트 통과 후 스테이징 브랜치를 프로덕션 브랜치로 병합하여 배포합니다.

B. 프롬프트를 MLflow 버전으로 추적하고, 테스트 통과 후 별칭을 사용하여 배포합니다.
C. CI 실행기의 JSON 파일에 프롬프트를 저장하고, 실행 시마다 프로덕션 프롬프트를 덮어씁니다.
D. 델타 테이블에 프롬프트를 저장하고, 배포 시마다 프로덕션 테이블을 덮어써 일관성을 유지합니다.

질문 **8**

목적: 상담원 사용 시나리오*(*앱, *Slack, Teams* 등*)*에 적합한 대화형 사용자 인터페이스를 개발하세요.

생성형 AI 엔지니어는 고객 지원 상담원이 내부 PDF를 기반으로 질문하고 답변을 받을 수 있는 Databricks 앱을 개발하고 있습니다. 요구 사항: 사용자는 회사 ID로 인증해야 하며, 앱은 브라우저에 장기 토큰을 노출하지 않고 Mosaic AI 에이전트 엔드포인트를 호출해야 하고, 답변 접근은 각 사용자의 권한을 준수해야 합니다.

어떤 접근 방식이 이러한 요구 사항을 충족합니까?

A. Databricks App 백엔드를 사용하여 앱 자격 증명으로 에이전트 엔드포인트를 호출하고 앱의 인증된 컨텍스트를 통해 사용자 신원/권한을 적용합니다.
B. Databricks 개인 액세스 토큰(PAT)을 앱의 JavaScript에 저장하고 브라우저에서 에이전트 엔드포인트를 직접 호출합니다.
C. 에이전트 엔드포인트를 공개적으로 게시하고 앱 프런트엔드에 내장된 API 키로 보호합니다.
D. 에이전트가 신원 확인 없이 PDF 파일을 읽을 수 있도록 PDF 파일을 공개 버킷으로 내보냅니다.

질문 **9**

목적: 주어진 애플리케이션 요구 사항에 따라 관리형, 외부 및 사용자 지정 *MCP* 서버를 통합합니다.

생성형 AI 엔지니어가 인터넷 데이터 소스에서 사실 정보에 접근하고 외부 API를 사용하여 웹 검색을 수행해야 하는 연구 보조 에이전트를 개발하고 있습니다. Databricks는 이 인터넷 데이터 소스에 대한 관리형 MCP 서버를 제공하고 있으며, 외부 API에는 키가 필요한 외부 MCP 서버가 있습니다. 애플리케이션은 운영 환경에서 두 데이터 소스 모두에 안정적으로 접근할 수 있도록 보장하면서 유지 관리 오버헤드를 최소화해야 합니다.

엔지니어는 이러한 데이터 소스를 에이전트에 통합하기 위해 어떤 두 가지 조치를 취해야 할까요?

A. 인터넷 리소스와 외부 API를 에이전트가 호출할 수 있는 단일 통합 인터페이스로 래핑하는 사용자 지정 MCP 서버를 구축합니다.
B. 관리형 웹 브라우저 MCP 서버를 사용하여 정보를 검색하기 위해 인터넷 리소스로

프로그래밍 방식으로 이동합니다.

C. 에이전트가 오프라인에서 액세스할 수 있도록 인터넷 리소스의 콘텐츠와 검색 결과를 캐시하도록 Unity Catalog 외부 테이블을 구성합니다.
D. 에이전트의 MCP 서버 구성에서 서버 유형을 "관리형"으로 지정하고 인터넷 리소스의 서버 식별자를 제공하여 관리형 MCP 서버를 구성합니다.
E. 외부 MCP 서버의 연결 정보를 제공하고 API 키를 Databricks Secrets에 저장한 다음 MCP 서버 구성에서 참조하여 외부 MCP 서버를 배포합니다.

질문 **10**

목적: 전문가 피드백을 반영하여 상담원 성능을 개선하세요

생성형 AI 엔지니어는 운영팀에서 내부적으로 사용하는 고객 지원 RAG(Real-Assistant Guide) 도우미를 평가하는 업무를 담당합니다. 네 명의 도메인 전문가가 매주 MLflow에서 샘플링된 답변을 검토하며, 사실성, 완전성, 유용성 등의 기준을 적용합니다. 여러 차례 검토를 거친 후, 엔지니어는 동일한 답변에 대한 전문가 평가가 크게 차이가 나는 것을 발견했으며, 이로 인해 모델 개선 추이를 추적하는 데 평가 데이터의 신뢰성이 떨어지는 것을 확인했습니다. 엔지니어는 평가자 간 불일치를 줄이면서 반복적인 품질 개선을 지원하는 신뢰할 수 있는 평가 프로세스를 구축해야 합니다.

엔지니어는 어떻게 해야 할까요?

A. LLM을 심사위원으로 활용하여 과거 및 미래 응답에 대한 점수를 재평가하고, 전문가 의견 불일치를 조정하는 대신 모델이 생성한 평가를 주요 기준으로 삼습니다.
B. 명확한 평가 기준을 정의하고, 해당 기준에 대해 전문가들을 교육한 후, **`mlflow.genai.evaluate()`** 함수에서 정렬된 판단 결과를 사용하여 일관된 에이전트 평가를 수행합니다.
C. 각 응답에 대해 모든 도메인 전문가의 점수를 평균하고, 이 평균 점수를 모델 튜닝을 위한 최종 벤치마크로 직접 사용합니다.
D. 모든 전문가가 이미 동의한 응답만을 사용하여 벤치마크를 구축하고, 일관성을 높이기 위해 평가 세트에서 의견이 다른 사례를 제외합니다.

답변

질문 1: A, B

질문 2: B

질문 3: C

질문 4: A

질문 5: D

질문 6: C

질문 7: B

질문 8: A

질문 9: D,E

질문 10: B