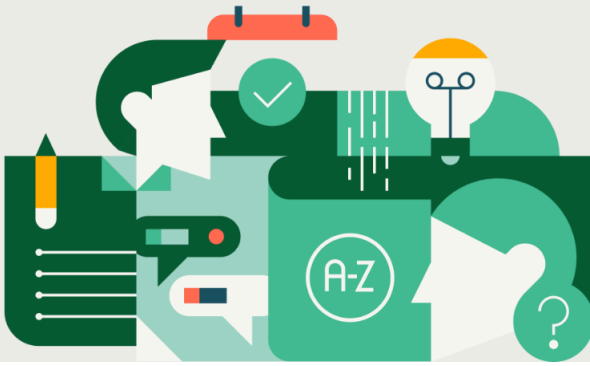
databricks

Databricks Exam 가이드

Databricks Certified Machine Learning Associate



시험 가이드 피드백 제공

정보 업데이트

업데이트	날짜
시험 출시	2025년 3월 1일
시험 가이드 업데이트	2025년 3월 1일

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

이 시험 가이드의 목적

이 시험 가이드의 목적은 시험 및 시험 범위에 대한 개요를 제공하여 시험 준비 여부를 판단하는 데 도움을 드리는 것입니다. 이 문서는 시험에 변경 사항이 있을 때마다 업데이트되며(변경 사항이 적용되는 시기도 함께 표시됨), 이를 통해 확신을 가지고 준비하실 수 있습니다. 이 가이드는 위에 명시된 발효일 기준으로 현재 운영 중인 시험 버전을 다룹니다.

대상 독자 설명

Databricks Certified Machine Learning Associate 자격증 시험은 Databricks를 사용하여 기본적인 머신 러닝 작업을 수행하는 개인의 능력을 평가합니다. 여기에는 Databricks와 AutoML, Unity Catalog, MLflow의 일부 기능 등 머신 러닝 역량을 이해하고 활용하는 능력이 포함됩니다. 또한 데이터를 탐색하고 특성 엔지니어링을 수행하는 능력도 평가합니다. 아울러 학습, 튜닝, 평가 및 선택을 통한 모델 구축 능력도 평가합니다. 마지막으로 머신 러닝 모델을 배포하는 능력을 평가합니다. 이 자격증 시험에 합격한 개인은 Databricks 및 관련 도구를 사용하여 기본적인 머신 러닝 작업을 완수할 수 있는 것으로 기대할 수 있습니다.

시험 소개

항목	세부 정보
채점 문항 수	채점되는 객관식 및 다중 선택 48문항
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 48개 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 또는 시험 센터
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**:** 인증 상태를 유지하려면 **[2]**년마다 재인증이 필요합니다**.** 재인증을 받으려면 현재 시행 중인 시험에 응시해야 합니다**.**

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

권장 준비 사항

학습

- 강사 주도**: Machine Learning with Databricks**
- 자기 주도 학습 **(Databricks Academy**에서 이용 가능**):**
    - 머신 러닝을 위한 데이터 준비
    - 머신 러닝 모델 배포
    - 머신 러닝 모델 개발

- 머신 러닝 운영

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 기반으로 프롬프트를 설정하고, AI가 제공하는 오래된 답변을 파악하며, 취약한 부분을 찾아내고, 모든 Databricks 시험에서 요구되는 실습으로 넘어갑니다.

실습으로 넘어갑니다

무엇을 연습해야 할지 모르시겠습니까? AI Prep 가이드(위의 AI Prep 가이드 섹션 참조)에는 "5단계 — 실습 최소 요건" 아래에 프롬프트가 포함되어 있으며, 이 프롬프트를 통해 AI 챗봇이 이 시험의 목표에 맞춰 6~10개의 실습 과제로 구성된 개인화된 체크리스트를 생성합니다. 각 과제는 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 있는지을 학습하고 있는지 확인하십시오. 공식 문서: Databricks 문서

시험 개요

섹션 1: Databricks 머신러닝 (38%)

- MLOps 전략의 모범 사례를 파악합니다
- ML 런타임 사용의 장점을 파악합니다
- AutoML이 모델/기능 선택을 어떻게 용이하게 하는지 파악합니다.
- AutoML이 모델 개발 프로세스에 가져다주는 이점을 파악합니다
- Databricks의 Unity Catalog에서 어카운트 수준으로 특성 저장소 테이블을 생성하는 것과 워크스페이스 수준으로 생성하는 것의 이점을 식별하십시오.
- Unity Catalog에 특성 저장소 테이블 생성
- 특성 저장소 테이블에 데이터를 씁니다.
- 특성 저장소 테이블의 특성을 사용하여 훈련합니다.
- 특성 저장소 테이블의 기능을 사용하여 모델을 스코어링합니다.
- 온라인 기능 테이블과 오프라인 기능 테이블의 차이점을 설명하십시오.
- MLflow Client API를 사용하여 최적의 실행을 식별합니다.
- MLflow 실행에서 메트릭, 아티팩트, 모델을 수동으로 로그합니다.

- MLflow UI에서 사용 가능한 정보를 식별합니다.
- Unity Catalog 레지스트리에서 MLflow Client API를 사용하여 모델을 등록합니다.
- 워크스페이스 레지스트리 대신 Unity Catalog 레지스트리에 모델을 등록할 때의 이점을 파악합니다.
- 코드 승격이 모델 승격보다 선호되는 시나리오와 그 반대의 경우를 식별합니다
- 모델의 태그 설정 또는 제거
- 별칭을 사용하여 챌린저 모델을 챔피언 모델로 승격합니다

Section 2: Data Processing (19%)

- .summary() 또는 dbutils 데이터 요약을 사용하여 Spark DataFrame에서 요약 통계를 계산합니다.
- 표준 편차 또는 IQR을 기반으로 Spark DataFrame에서 이상값을 제거합니다.
- 범주형 또는 연속형 특성에 대한 시각화 생성
- 적절한 방법을 사용하여 두 개의 범주형 특성 또는 두 개의 연속형 특성을 비교합니다.
- 평균, 중앙값 또는 최빈값으로 결측값을 대체하는 방법을 비교하고 대조합니다.
- 최빈값, 평균값 또는 중앙값으로 결측값을 대체합니다.
- 범주형 특성에 원-핫 인코딩을 사용합니다
- 원-핫 인코딩이 적합하거나 적합하지 않은 모델 유형 또는 데이터 세트를 식별하고 설명합니다.
- 로그 스케일 변환이 적절한 시나리오를 식별합니다

Section 3: Model Development (31%)

- ML 기초를 활용하여 주어진 모델 시나리오에 적합한 알고리즘을 선택합니다.
- 학습 데이터의 데이터 불균형을 완화하는 방법 식별
- 추정기와 변환기 비교
- 학습 파이프라인 개발
- Hyperopt의 fmin 연산을 사용하여 모델의 하이퍼파라미터를 튜닝합니다
- 하이퍼파라미터 튜닝 방법으로 랜덤 탐색, 그리드 탐색 또는 베이지안 탐색을 수행합니다.
- 하이퍼파라미터 튜닝을 위해 단일 노드 모델을 병렬화합니다
- 훈련-검증 분할 대신 교차 검증을 사용할 때의 장점과 단점을 설명하십시오.
- 모델 피팅의 일부로 교차 검증을 수행합니다.
- 그리드 검색 및 교차 검증 프로세스와 함께 학습되는 모델의 수를 파악합니다.
- 일반적인 분류 메트릭(F1, Log Loss, ROC/AUC 등)을 사용합니다.
- 일반적인 회귀 메트릭(RMSE, MAE, R-squared 등)을 사용합니다.
- 주어진 시나리오 목표에 가장 적합한 지표를 선택합니다
- 평가 메트릭을 계산하거나 예측을 해석하기 전에 로그 변환된 변수를 지수 변환해야 할 필요성을 파악합니다
- 모델 복잡성과 편향-분산 트레이드오프가 모델 성능에 미치는 영향을 평가합니다

섹션 4: 모델 배포 (12%)

- 배치, 실시간, 스트리밍 등 Model Serving 방식의 차이점과 장점을 파악합니다.
- 커스텀 모델을 모델 엔드포인트에 배포합니다

- Pandas를 사용하여 배치 추론을 수행합니다
- Delta Live Tables를 사용하여 스트리밍 추론을 수행하는 방법을 파악합니다.
- 실시간 추론을 위한 모델 배포 및 쿼리
- 실시간 추론을 위해 엔드포인트 간에 데이터를 분할합니다.

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 위에 있는 시험 날짜에 해당하는 전체 시험 개요를 검토하는 것입니다.

문제 1

목표***: Unity Catalog***에 특성 저장소 테이블을 생성합니다**.**

데이터 과학자가 모델에서 사용할 특성 테이블을 생성하려고 합니다. 해당 데이터 과학자는 Unity Catalog가 활성화된 워크스페이스에서 작업 중이며, 이 특성 테이블을 Unity Catalog에 저장하고 관리하려고 합니다. 이 특성 테이블을 생성하는 올바른 방법은 무엇입니까?

A. 평소와 같이 데이터가 포함된 Delta 테이블을 생성한 다음, Python의 FeatureStoreClient에서 register_table 메서드를 사용하여 Unity Catalog에 특성 테이블로 등록합니다.
B. SQL을 통해 AS FEATURE STORE 절을 사용하여 Unity Catalog에 빈 Delta 테이블을 생성한 다음, 데이터를 작성합니다.
C. Python에서 FeatureEngineeringClient의 create_table 메서드를 사용하여 테이블을 생성한 다음, 데이터를 해당 테이블에 씁니다.
D. Unity Catalog에 데이터가 포함된 Delta 테이블을 생성한 다음, SQL에서 ALTER TABLE 명령을 사용하여 SET AS FEATURE STORE 절로 해당 테이블을 특성 테이블로 구성합니다.

질문 2

목표**:** 최빈값**,** 평균값 또는 중앙값으로 결측값을 대체합니다**.**

데이터 과학자가 연속형 특성의 결측값을 대체해야 합니다. 최소한의 노력으로 올바른 결과를 얻고자 합니다. 어떤 전략이 이를 수행할 수 있습니까?

A. sklearn SimpleImputer를 사용하면 기능 분포에 따라 최적의 방법론을 자동으로 선택합니다.
B. 값의 분포를 검토하고 검토 결과에 따라 적절한 대체(imputation) 방법을 선택합니다.
C. 연속형 열에 가장 적합한 대체 방법인 .mean()을 사용합니다.
D. 연속형 열에 가장 적합한 대체 방법인 .mode()를 사용합니다.

문제 3

목표**:** 학습 데이터의 데이터 불균형을 완화하는 방법을 식별합니다**.**

데이터 과학자가 고객이 구독 서비스를 이탈할지 여부를 예측하는 모델을 개발하고 있습니다. 데이터셋은 심각하게 불균형하여, 전체 인스턴스 중 해지하는 고객을 나타내는 경우가 10%에 불과합니다. 데이터 과학자는 모델이 다수 클래스에 편향되지 않으면서 소수 클래스를 효과적으로 식별하도록 하고자 합니다. 클래스 불균형으로 인한 비해지 고객 방향의 모델 편향을 직접적으로 완화하는 전략은 무엇입니까?

A. 특성을 정규화하여 동일한 스케일을 갖도록 함으로써 모델 성능을 향상시킵니다.
B. 모델 학습 중 소수 클래스에 더 높은 오분류 비용을 할당하여 비용 민감 학습을 사용합니다.
C. 이탈하지 않은 고객에 대한 데이터를 추가로 수집하여 학습 데이터셋의 크기를 늘립니다.
D. 더 단순한 모델을 사용하여 과적합을 줄이고, 소수 클래스에 대한 일반화 성능을 향상시킵니다.

문제 4

목표**:** 그리드 서치 및 교차 검증 프로세스와 함께 학습되는 모델의 수를 파악합니다**.**

데이터 과학자가 scikit-learn의 5-fold 교차 검증과 GridSearchCV를 사용하여 Support Vector Machine(SVM) 모델을 튜닝하고 있습니다. 파라미터 그리드에는 최적화할 세 가지 하이퍼파라미터가 포함되어 있습니다: 값이 [0.1, 1, 10]인 C, 선택지가 ['linear', 'rbf']인 kernel, 값이 [0.01, 0.1, 1]인 gamma. 총 몇 개의 서로 다른 모델이 학습됩니까?

A. 90
B. 18
C. 1
D. 위의 어느 것도 해당하지 않습니다.

질문 5

목표***: Delta Live Tables***를 사용하여 스트리밍 추론을 수행하는 방법을 식별합니다**.**

한 회사가 수천 명의 사용자를 보유한 팟캐스트 플랫폼을 운영하고 있습니다. 이 회사는 청취, 일시 정지, 팟캐스트 종료 등의 사용자 이벤트에 대한 10분 단위 롤링 윈도우를 기반으로 팟캐스트 참여도 저하를 감지하는 이상 탐지 알고리즘을 구현했습니다. 한 머신 러닝 엔지니어가 초당 수만 건의 이벤트를 처리해야 하는 프로덕션 데이터 파이프라인에 이 모델을 배포하려고 합니다. 하루 동안 이벤트 볼륨이 변동함에 따라, 엔지니어는 파이프라인 compute가 동적으로 크기 조정될 수 있어야 합니다. 이러한 요구 사항을 충족하는 파이프라인 설계 방식은 무엇입니까?

A. 알고리즘을 Spark UDF로 적용하는 Delta Live Tables 파이프라인을 생성합니다.
B. 알고리즘을 Spark UDF로 적용하는 Structured Streaming 작업을 생성합니다.
C. model serving 엔드포인트를 생성하고, 해당 엔드포인트를 호출하는 커스텀 UDF를 사용하는 Delta Live Tables 파이프라인을 생성합니다.
D. Model Serving 엔드포인트를 생성하고, 해당 엔드포인트를 호출하는 사용자 정의 UDF를 실행하는 Structured Streaming 작업을 생성합니다.

질문 6

목표**:** 별칭***(alias)***을 사용하여 챌린저 모델을 챔피언 모델로 승격합니다**.**

팀이 Unity Catalog model registry에 모델의 연속적인 버전을 등록합니다. 새로 등록된 버전이 현재 프로덕션 트래픽을 처리하는 버전보다 성능이 우수하다는 검증 결과가 나온 후, 권장 방법에 따라 새 버전을 생산에서 읽는 버전으로 표시하려고 합니다. 어떻게 해야 합니까?

A. MlflowClient().transition_model_version_stage()를 사용하여 새 버전을 "Production" 단계로 이동합니다.
B. MlflowClient().set_registered_model_alias()를 사용하여 새 모델 버전에 "champion" 별칭을 할당합니다.
C. 현재 챔피언 모델 버전을 삭제하여 새 버전만 서빙에 사용할 수 있도록 합니다.
D. 새 모델 버전에 "champion" = "true" 태그를 설정하려면 MlflowClient().set_model_version_tag()를 사용합니다.

질문 7

목표***: MLflow Client API***를 사용하여 최적의 실행을 식별합니다**.**

한 데이터 과학자가 단일 실험에 수십 개의 MLflow 실행을 기록했으며, 각 실행에는 validation_rmse 메트릭이 기록되어 있습니다. 이 데이터 과학자는 MLflow Client API를 사용하여 프로그래밍 방식으로 validation_rmse가 가장 낮은 단일 실행을 검색하려고 합니다. 이를 수행하는 방법은 무엇입니까?

A. 실험 ID와 함께 `search_runs`를 호출하고, `order_by=["metrics.validation_rmse ASC"]`를 지정한 후 첫 번째로 반환된 실행을 가져옵니다.
B. MLflow UI에서 실험을 열고, 실행 테이블을 validation_rmse 기준으로 정렬한 후 맨 위 행을 확인합니다.
C. 클라이언트로 실행 목록을 조회하고 가장 최근 시작 시간을 가진 실행을 선택합니다.
D. 실험 이름을 사용하여 get_run을 호출하면 실험의 최적 실행을 직접 반환합니다.

문제 8

목표**:** 로그 스케일 변환이 적절한 시나리오를 식별합니다**.**

데이터 과학자가 주택 가격 회귀 모델을 위한 기능을 준비하고 있습니다. 하나의 수치형 기능과 가격 타겟 모두 강하게 오른쪽으로 치우쳐 있으며, 여러 자릿수에 걸쳐 있는 매우 큰 값이 소수 존재합니다. 로그 스케일 변환을 적용하는 것이 가장 적절한 상황은 언제입니까?

A. 범주형 기능에 고유 수준이 많고 카디널리티를 줄여야 할 때.
B. 기능이 이미 근사적으로 정규 분포를 따르며 0에 가깝게 중심화되어 있을 때.
C. 기능 또는 타겟이 오른쪽으로 치우쳐 있고 여러 자릿수에 걸쳐 있을 때, 큰 값을 압축하고 분포를 더 대칭적으로 만들기 위해 사용합니다.
D. 기능에 양수 범위로 이동해야 하는 음수 값이 포함된 경우.

질문 9

목표**:** 모델 복잡성과 편향**-**분산 트레이드오프가 모델 성능에 미치는 영향을 평가합니다**.**

평가 중에 데이터 과학자는 자신의 모델이 학습 세트에서는 매우 낮은 오류를 달성하지만 검증 세트에서는 상당히 높은 오류를 보인다는 것을 관찰합니다. 편향-분산 트레이드오프를 기반으로, 어떤 현상이 발생하고 있으며 적절한 대응 방법은 무엇입니까?

A. 모델에 높은 편향(과소적합)이 있으므로, 데이터 과학자는 모델 복잡도를 높여야 합니다.
B. 모델은 너무 적은 특성으로 인해 높은 편향이 발생하고 있으며, 데이터 과학자는 추가 특성을 엔지니어링해야 합니다.
C. 학습 오류와 검증 오류 모두 높으며, 이는 높은 분산을 나타냅니다. 과학자는 모델을 더 오래 훈련해야 합니다.
D. 모델의 분산이 높습니다(과적합). 데이터 과학자는 모델 복잡도를 줄이거나 정규화를 추가해야 합니다.

질문 10

목표**:** 배치**,** 실시간**,** 스트리밍 등 ***model serving*** 방식의 차이점과 장점을 파악합니다**.**

한 소매업체가 다음 날의 마케팅 세그먼트를 생성하기 위해 매일 밤 한 번씩 전체 고객 기반에 대해 점수를 산출합니다. 낮은 지연 시간의 요청별 예측에 대한 요구 사항은 없습니다. 이 워크로드에 가장 적합하고 비용 효율적인 Model Serving 방식은 무엇입니까?

A. 매일 밤 일정에 따라 모든 레코드를 스코어링하는 배치 추론입니다.
B. 고객당 한 번씩 쿼리되는 REST model serving 엔드포인트를 통한 실시간 추론입니다.
C. 지속적으로 실행되는 Structured Streaming 파이프라인을 사용한 스트리밍 추론.
D. 가능한 가장 낮은 지연 시간을 위해 크기가 설정된 상시 가동 GPU model serving 엔드포인트입니다.

정답 키 및 해설

#	정답	설명
1	C	Unity Catalog가 활성화된 워크스페이스에서는 **`FeatureEngineeringClient.create_table`** 메서드로 피처 테이블을 생성한 뒤 데이터를 씁니다. **`AS FEATURE STORE`** SQL 절도, **`ALTER TABLE … SET AS FEATURE STORE`** 구문도 존재하지 않으며, **`register_table`**도 현재의 생성 방법이 아닙니다.
2	B	올바른 결과를 얻으려면 피처의 분포를 살펴보고 적절한 대치(imputation) 방법을 선택해야 합니다. **`SimpleImputer`**는 분포로부터 전략을 자동으로 선택하지 않으며, 평균(또는 최빈값)을 무작정 적용하는 것은 특정 연속형 피처에 부적절할 수 있습니다.
3	B	비용 민감 학습(cost-sensitive learning)은 소수(이탈) 클래스에 더 높은 오분류 비용을 부여하여 다수 클래스로 치우치는 모델의 편향을 직접 상쇄합니다. 피처 정규화, 다수 클래스 데이터 추가 수집, 더 단순한 모델 사용은 클래스 불균형을 해결하지 못합니다.
4	A	그리드에는 **`3 x 2 x 3 = 18`**개의 하이퍼파라미터 조합이 있으며, 5-폴드 교차 검증은 각 조합마다 폴드별로 모델을 학습하므로 총 **`18 x 5 = 90`**개의 모델이 학습됩니다.
5	A	알고리즘을 Spark UDF로 적용하는 Delta Live Tables 파이프라인은 높은 이벤트 처리량을 처리하고 파이프라인 컴퓨팅을 동적으로 조정합니다. 모델 서빙 엔드포인트를 거치면 오버헤드가 추가되고, Structured Streaming 잡은 동일한 관리형 동적 자동 확장을 제공하지 않습니다.
6	B	Unity Catalog 모델 레지스트리에서 승격 워크플로는 레거시 스테이지 전환 대신 별칭(alias)을 사용합니다. 새 버전에 **`"champion"`** 별칭을 **`set_registered_model_alias`**(으)로 지정하면 해당 버전이 운영 환경에서 읽는 버전이 됩니다. 스테이지 전환은 더 이상 사용되지 않는 워크스페이스 레지스트리에 속하고, 이전 버전을 삭제하는 것은 파괴적이고 불필요하며, 태그는 서빙되는 버전을 바꾸지 않는 설명용 메타데이터입니다.
7	A	MLflow 클라이언트의 **`search_runs`**(예: **`mlflow.search_runs`**)는 서버 측 정렬을 지원하므로, **`metrics.validation_rmse`**을(를) 오름차순으로 정렬하고 첫 번째 결과를 가져오면 프로그래밍 방식으로 최적의 실행을 반환합니다. MLflow UI는 클라이언트 API가 아니고, 가장 최근에 시작된 실행이 반드시 최적인 것은 아니며, **`get_run`**은(는) 특정 **`run_id`**을(를) 필요로 하며 실행 순위를 매기지 않습니다.
8	C	로그 변환은 큰 값을 압축하고 긴 오른쪽 꼬리를 끌어당겨, 여러 자릿수에 걸쳐 분포하는 오른쪽으로 치우친(right-skewed) 피처나 타깃을 더 대칭적으로 만듭니다. 이는 카디널리티 축소 기법이 아니며, 이미 정규 분포인 피처에는 이점이 없고, 0이나 음수 값에 대해서는 정의되지 않으므로 음수를 양수 범위로 옮기는 데 사용되지 않습니다.
9	D	낮은 학습 오차와 함께 훨씬 높은 검증 오차가 나타나는 것은 높은 분산, 즉 과적합(overfitting)의 전형적인 징후입니다. 적절한 대응은 모델 복잡도를 줄이거나 정규화를 추가하는 것(또는 더 많은 학습 데이터를 수집하는 것)입니다. 높은 편향/과소적합(underfitting)이라면 대신 학습 세트와 검증 세트 모두에서 높은 오차가 나타납니다.
10	A	저지연 요구 사항 없이 전체 데이터셋에 대해 고정된 일정으로 예측을 생성하는 경우, 배치 추론(batch inference)이 가장 간단하고 비용 효율적인 방식입니다. 실시간 및 상시 가동 GPU 엔드포인트는

		워크로드에 필요하지 않은 지연 시간을 위해 인프라와 비용을 추가하며, 스트리밍 추론은 야간 대량 스코어링 잡이 아니라 지속적으로 도착하는 데이터를 대상으로 합니다.