

Machine Learning Operation(MLOps)

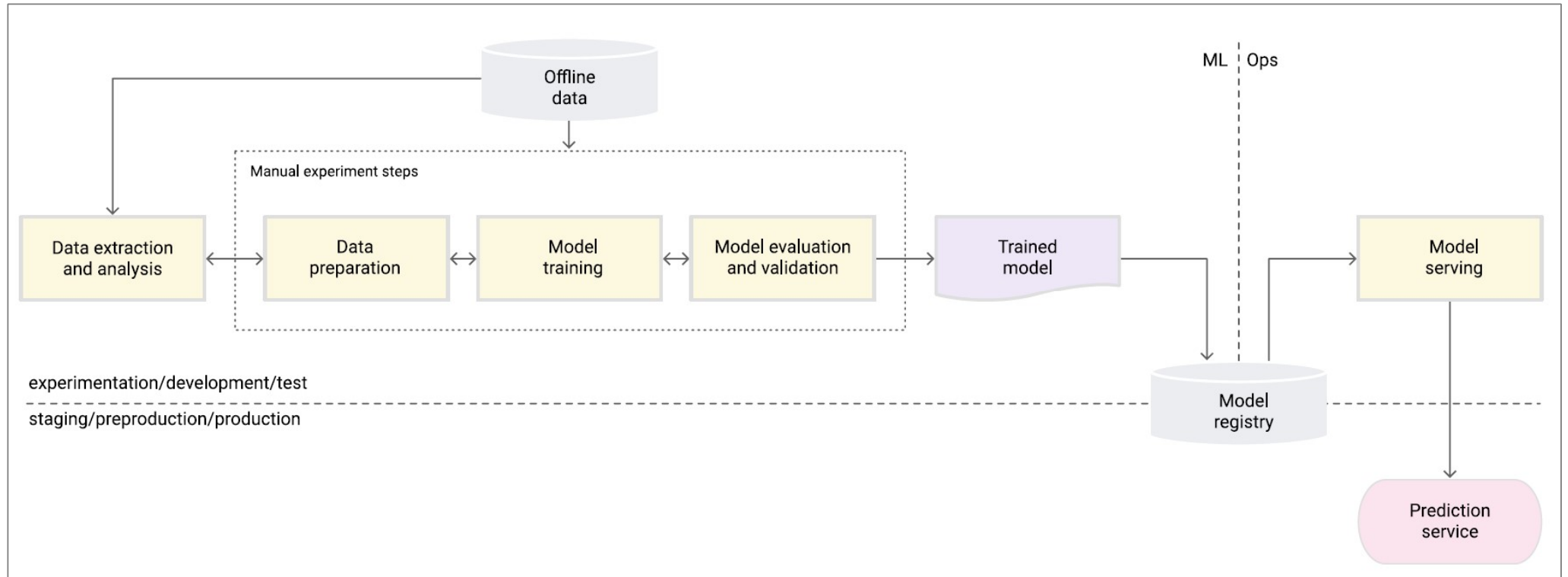
– Ch3. Level1 MLOps(1)

Level1 MLOps(1)

1. Level 0 MLOps 리뷰
2. Level 1 MLOps 특징

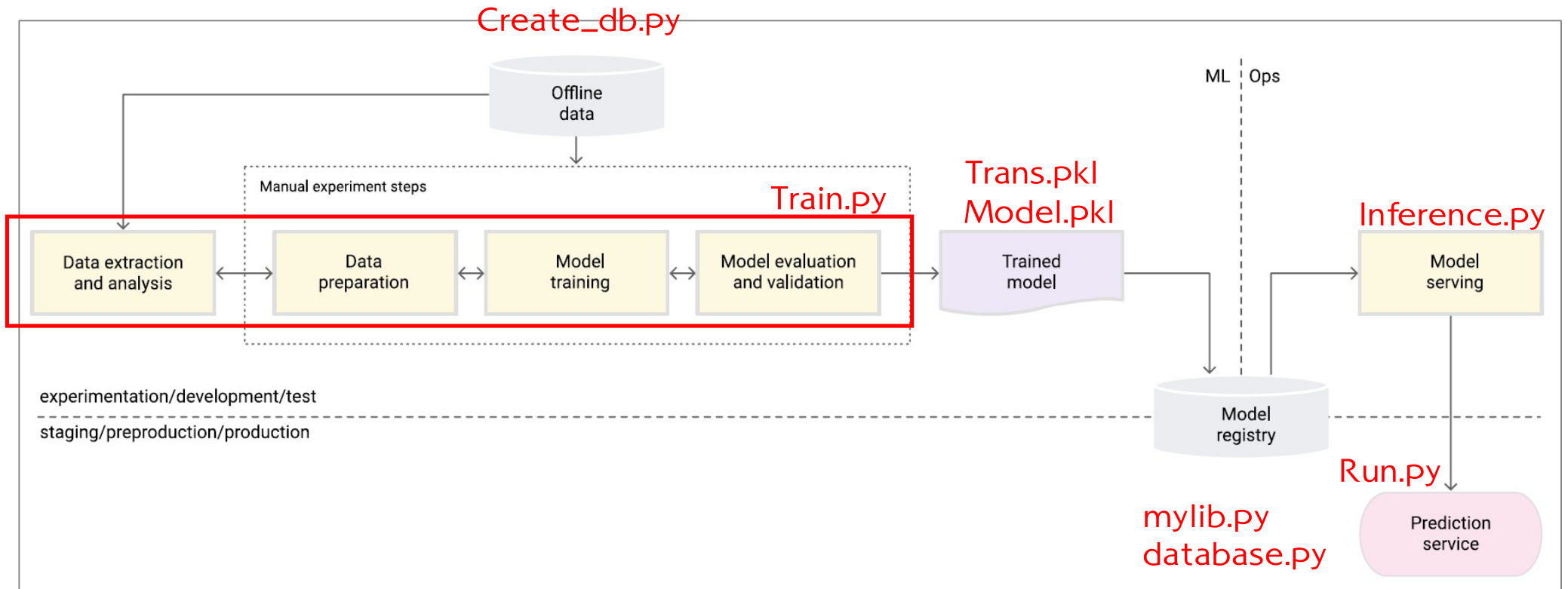
Level 0 MLOps 리뷰

Level 0 MLOps



[그림] 모델을 예측 서비스로 제공하기 위한 수동 ML 단계

Level 0 MLOps

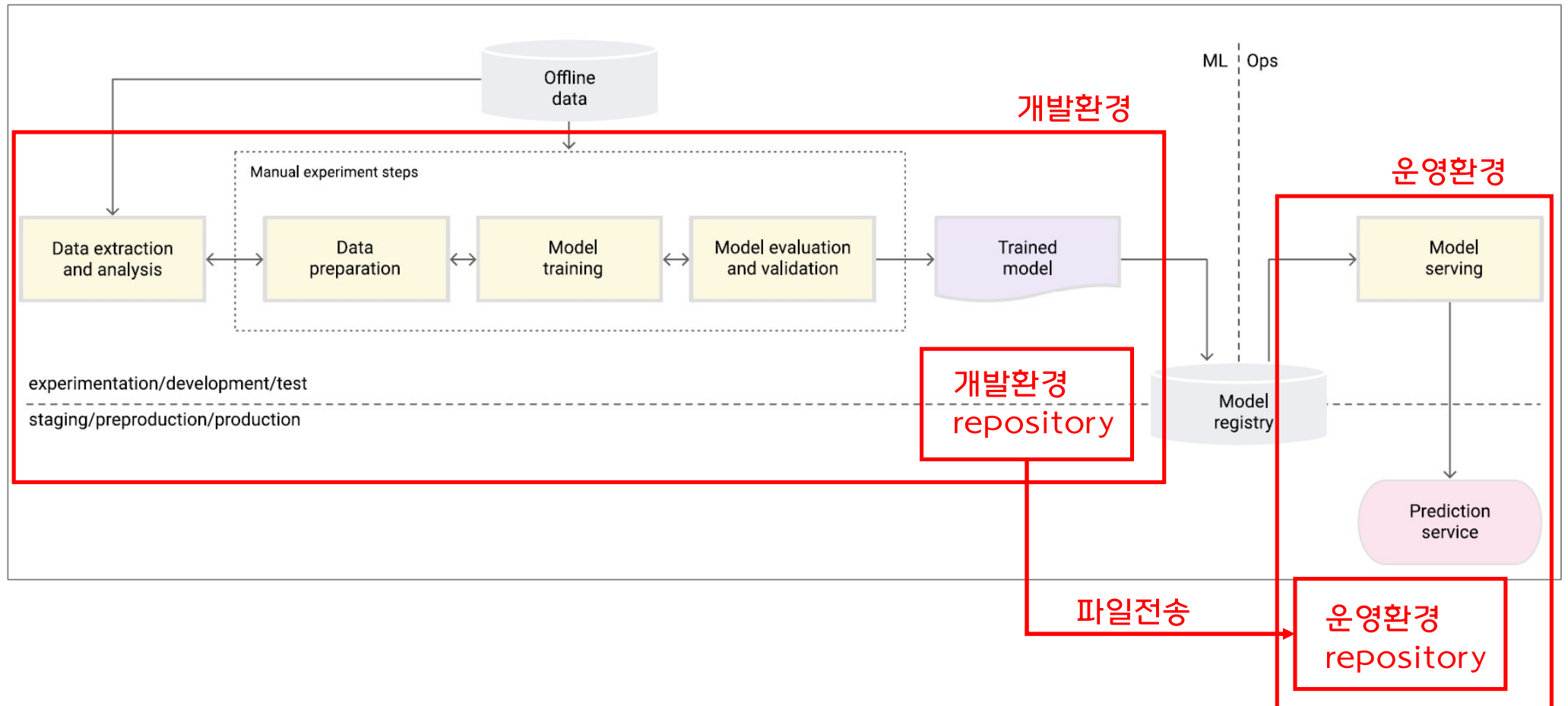


[그림] 모델을 예측 서비스로 제공하기 위한 수동 ML 단계

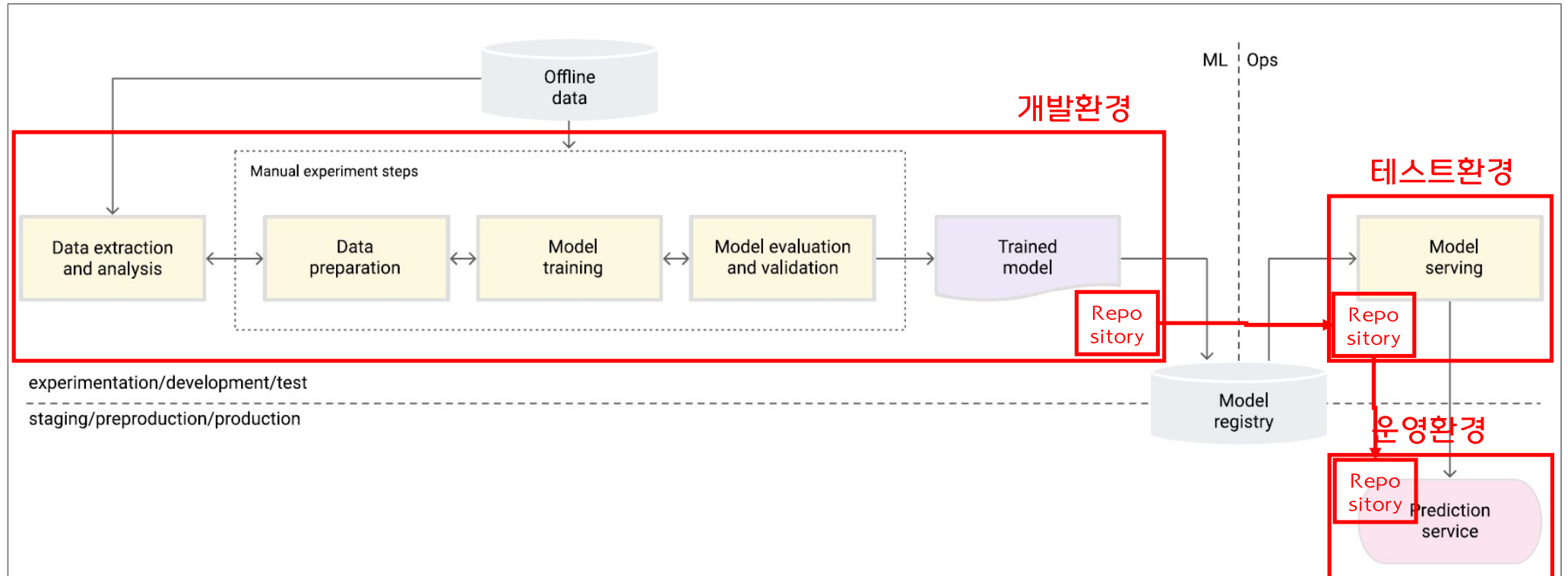
Level 0 MLOps

- **Manual script-driven, Interactive process:**
데이터 분석, 데이터 준비, 모델 학습, 모델 검증을 포함한 **모든 단계가 수동**.
각 단계를 수동으로 실행하고, 한 단계에서 다른 단계로 수동 전환.
이 프로세스는 일반적으로 작업 가능한 모델이 생성될 때까지 데이터 과학자가 노트북에서 작성하고 실행하는 실험 코드에 의해 구성됨
- **Disconnect between ML and Ops:**
모델을 만드는 데이터 과학자와 모델을 예측 서비스로 제공하는 엔지니어를 분리함
데이터 과학자는 엔지니어링팀에 학습된 모델을 아티팩트로 전달.
이 전달에는 학습된 모델을 스토리지 위치에 배치하거나, 모델 객체를 코드 저장소에 확인하거나, **모델 레지스트리에 업로드하는 작업이 포함**될 수 있습니다. 그런 다음 모델을 배포하는 엔지니어는 짧은 지연 시간을 제공하기 위해 필요한 기능을 프로덕션 단계에서 사용할 수 있도록 해야 함. 이를 통해 학습-서빙 편향이 발생할 수 있음
(https://developers.google.com/machine-learning/guides/rules-of-ml/?hl=ko#training-serving_skew)
- **Infrequent release iterations:**
데이터 프로세스는 데이터 과학팀이 모델 구현을 변경하거나 새 데이터로 모델을 재학습시키는 등 자주 변경되지 않는 몇 가지 모델을 관리한다고 가정함. **새 모델 버전은 1년에 두어 번만 배포**함.
- **No CI:**
구현 변경사항이 거의 없으므로 CI가 무시됨. 일반적으로 코드 테스트는 노트북 또는 스크립트 실행의 일부임. 실험 단계를 구현하는 스크립트와 노트북은 소스로 제어되며 학습된 모델, 평가 측정항목, 시각화와 같은 아티팩트를 생성함.
- **No CD:**
모델 버전이 자주 배포되지 않으므로 CD는 고려되지 않음.
- **Deployment refers to the prediction service:**
이 프로세스는 전체 ML 시스템을 배포하는 대신 **학습된 모델을 예측 서비스로 배포**하는 것에만 관여함
- **Lack of active performance monitoring:**
프로세스는 모델 성능 저하 및 기타 모델 동작 드리프트를 감지하는 데 필요한 모델 예측 및 작업을 **추적하거나 기록하지 않음**.

Level 0 MLOps

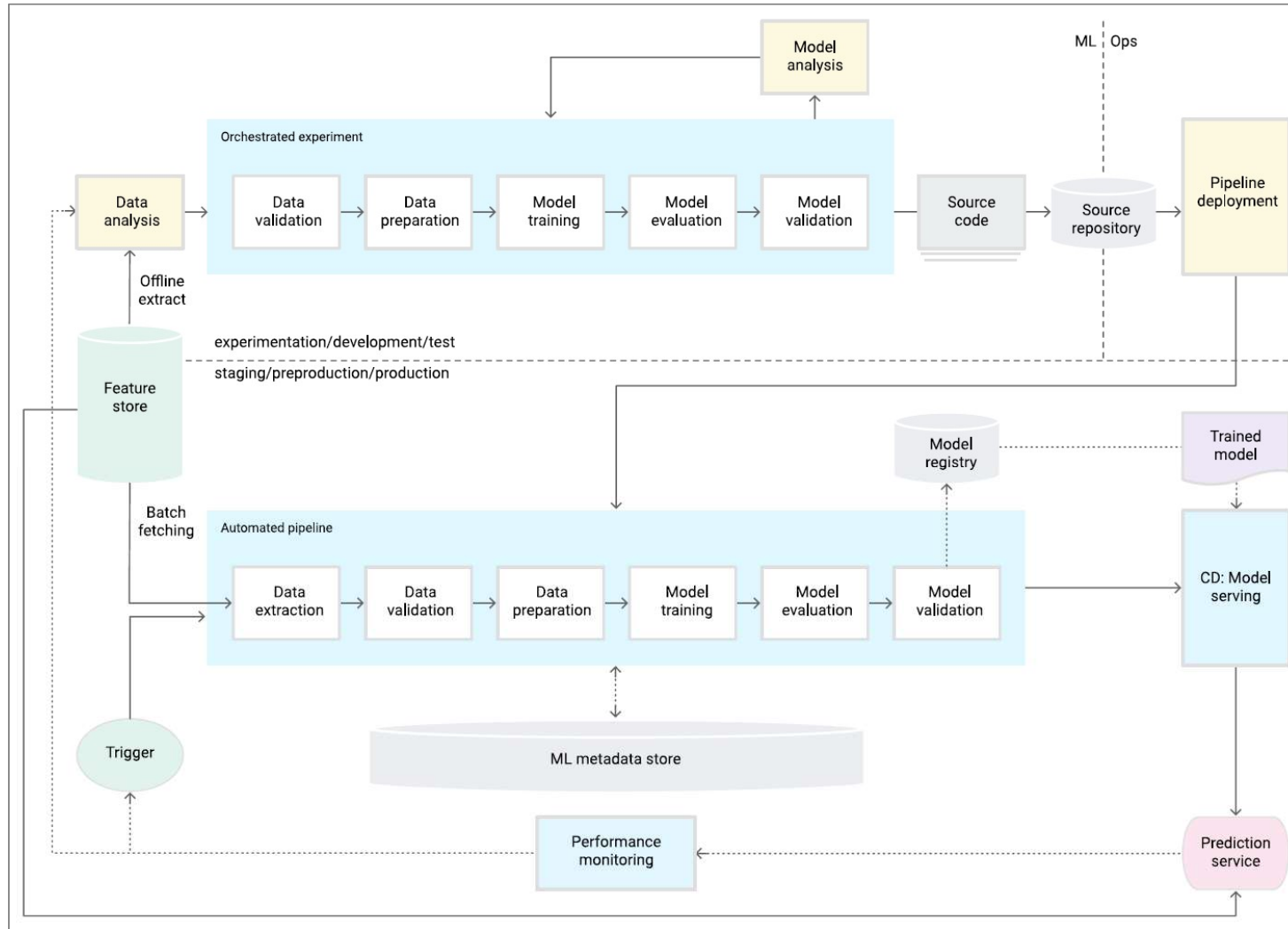


Level 0 MLOps



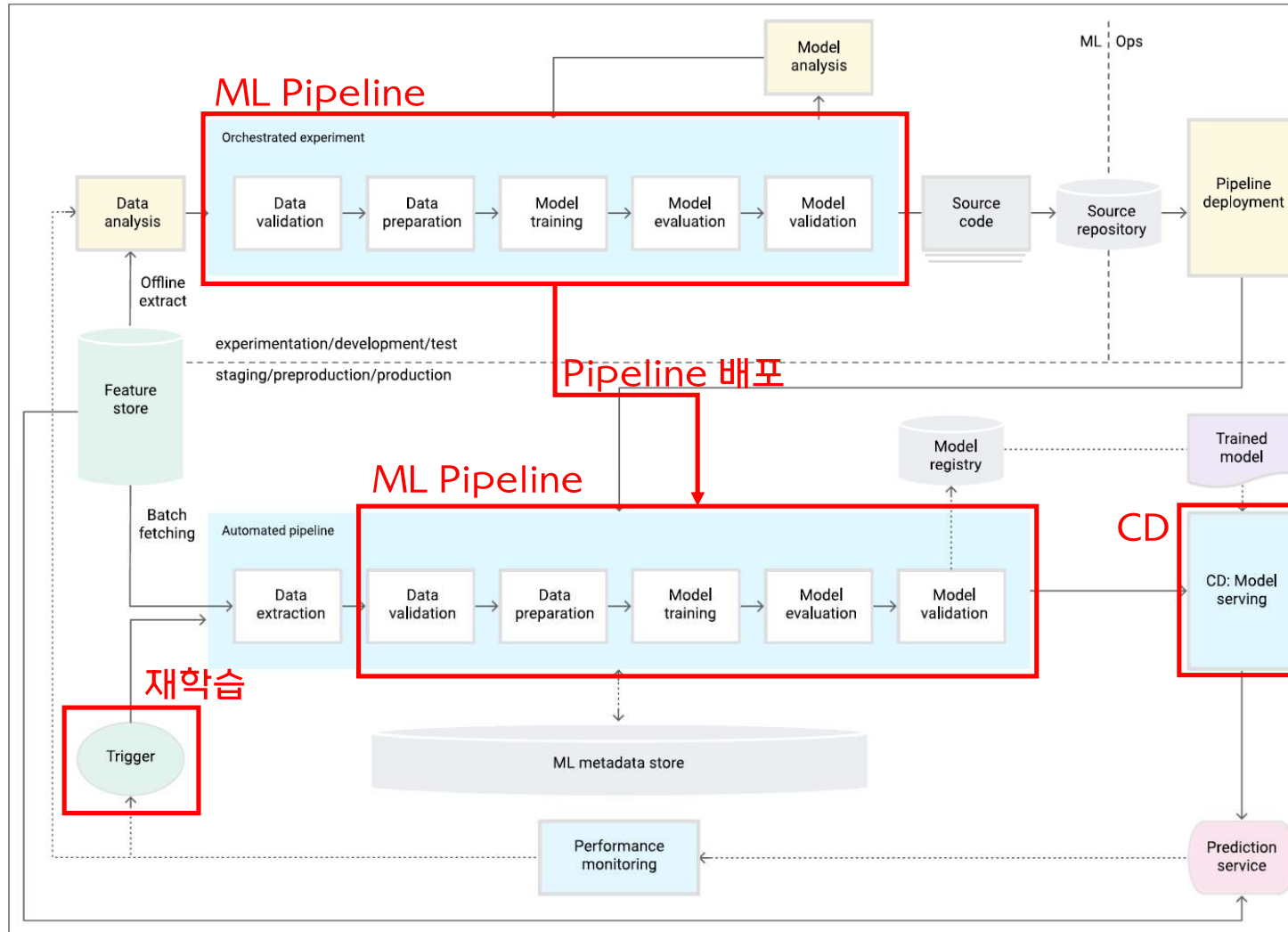
Level 1 MLOps 특징

Level 1 MLOps



[그림] CT용 ML 파이프라인 자동화

Level 1 MLOps



[그림] CT용 ML 파이프라인 자동화

Level 1 MLOps

- **빠른 실험:**
ML 실험 단계가 조정됨. 단계 간 전환은 자동으로 이루어지므로 실험을 빠르게 반복하고, 전체 파이프라인을 프로덕션으로 더 빠르게 이동할 수 있음.
- **프로덕션 단계에서 모델의 CT:**
다음 섹션에서 설명하는 실시간 파이프라인 트리거를 기반으로 하는 새로운 데이터를 사용하여 모델이 **프로덕션 단계에서 자동으로 학습**함.
- **실험-운영 균형:**
개발 또는 실험 환경에서 사용되는 파이프라인 구현은 사전 프로덕션 및 프로덕션 환경에서 사용되며, 이는 DevOps 통합을 위한 MLOps 방식에 있어 핵심적인 요소임.
- **구성요소 및 파이프라인의 모듈화된 코드:**
ML 파이프라인을 구성하려면 ML 파이프라인 전체에서 구성요소를 재사용, 구성, 공유할 수 있어야 함. 따라서 EDA 코드는 여전히 노트북에 있을 수 있지만 구성요소의 **소스 코드는 모듈화**되어야 함. 또한 구성요소를 컨테이너화하여 다음을 수행하는 것을 권장함.
 - 커스텀 코드 런타임에서 실행 환경을 분리.
 - 개발 및 프로덕션 환경 간에 코드를 재현 가능하도록 함.
 - 파이프라인의 각 구성요소를 격리. 구성요소는 자체 런타임 환경 버전을 가질 수 있으며 다양한 언어 및 라이브러리를 가짐.
- **모델의 지속적 배포:**
프로덕션 단계의 ML 파이프라인은 새 데이터로 학습된 새 모델에 예측 서비스를 지속적으로 배포함. 학습 및 검증된 모델을 온라인 예측용 예측 서비스로 제공하는 **모델 배포단계가 자동화**됨.
- **파이프라인 배포:**
수준 0에서는 학습된 모델을 프로덕션에 예측 서비스로 배포함. 수준 1의 경우, 학습된 모델을 예측 서비스로 제공하기 위해 자동으로 반복 실행되는 **전체 학습 파이프라인을 배포**함.

End of Document