머신러닝 파이프라인

머신러닝 파이프라인의 이해

송호연

목차

머신러닝 파이프라인의 이해

학습목표

머신러닝 파이프라인의 이해

01. 머신러닝 파이프라인에 대해 이해한다.

머신러닝 파이프라인의 개념에 대해 이해한다.

02. 머신러닝 문제의 특징을 이해한다.

머신러닝 기반의 소프트웨어가 갖고 있는 특징을 이해한다.

03. MLOps의 핵심 문제에 대해 이해한다.

MLOps에서 풀어야 하는 핵심 문제에 대해 이해한다.

04. MLOps 성숙도 레벨에 대해 이해한다.

MLOps 성숙도 레벨에 따른 특징을 이해한다.

머신러닝 파이프라인의 필요성

헨리 포드의 자동차 조립 라인

파이프라인을 통해 얻는 이점

머신러닝 파이프라인의 필요성

01

머신러닝 파이프라인의 이해

헨리 포드의 자동차 조립 라인

헨리 포드의 회사가 1913년에 포드의 전설적인 모델 T를 생산하기 위해 첫 조립 라인을 만들었습니다. 조립 라인을 도입함으로 인해 각각의 자동차를
만드는 데 걸리는 시간을 12시간에서 3시간으로 줄였습니다. 조립 시간은 급격히 감소했습니다. 이렇게 해서 비용이 대폭 절감되었고, 모델 T는 역사상
최초의 저렴한 자동차가 될 수 있었습니다. 또한 대량 생산을 가능하게

출처 : https://ko.wikipedia.org/wiki/%ED%8F%AC%EB%93%9C_%EB%AA%A8%EB%8D%B8_T

머신러닝 파이프라인의 이해

헨리 포드의 자동차 조립 라인

생산 공정이 "잘 정의된 프로세스"(다른 말로 파이프라인)로 정립되었기 때문에
이러한 프로세스 중 일부를 자동화할 수 있게 되어 시간과 비용이 훨씬
절감되었습니다.
오늘날, 자동차는 대부분 기계에 의해 만들어집니다.

출처 : https://ko.wikipedia.org/wiki/%ED%8F%AC%EB%93%9C_%EB%AA%A8%EB%8D%B8_T

머신러닝 파이프라인의 이해

파이프라인을 통해 얻는 이점

생산성 향상

예측 가능한 품질

장애 대응능력

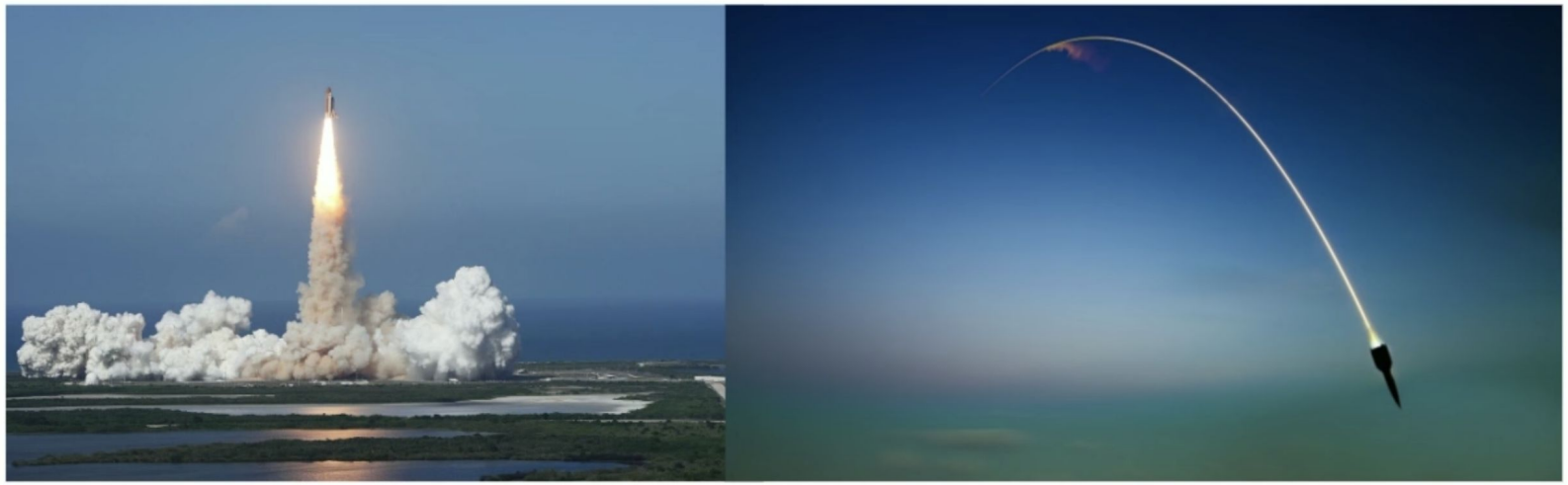
향상

머신러닝 파이프라인의 이해

머신러닝 파이프라인의 필요성

ML 관련 업무가 수년간 경험이 쌓이고 있다 보니
실제 시스템에 점점 통합이 되어가고 있는 추세입니다.

대부분의 회사에서 머신러닝 모델의 배포는 쉬워졌지만,
머신러닝 모델을 운영하기가 힘들다는 사실을 깨닫기
시작했습니다.



머신러닝 파이프라인의 이해

머신러닝 파이프라인의 필요성

ML 시스템 개발 및 배포는 비교적 쉽고 빠르지만 해당 시스템을 유지하고 관리하는 비용은 매우 큽니다.

머신러닝 파이프라인의 이해

머신러닝 파이프라인의 필요성

기술 부채
(Technical Debt)

- Ward Cunningham, 1992

머신러닝 파이프라인의 이해

머신러닝 파이프라인의 필요성

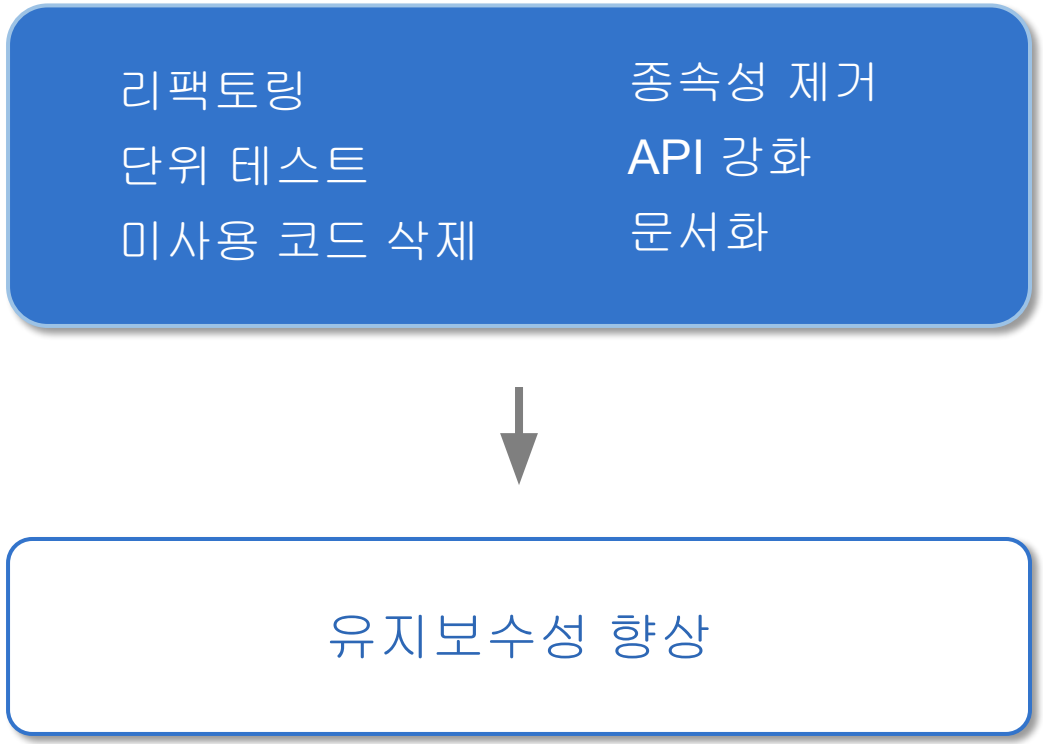
기술 부채
(Technical Debt)

- Ward Cunningham, 1992

- 리팩토링
- 단위 테스트
- 미사용 코드 삭제
- 종속성 제거
- API 강화
- 문서화

머신러닝 파이프라인의 이해

머신러닝 파이프라인의 필요성



머신러닝 파이프라인의 이해

머신러닝 파이프라인의 필요성

소프트웨어 공학에서 이야기하는 **추상화(Abstraction)**의
경계가
ML 시스템에서는 미묘하게 무너지게 됩니다.
기존에 알려진 코드 수준의 기술 부채 제거 방식으로는
이 문제를 해결하기 어렵습니다.

머신러닝 문제의 특징

쉬운/어려운 머신러닝 문제

머신러닝 프로그래밍 문제의 특징

02

머신러닝 문제의 특징

쉬운/어려운 머신러닝 문제

쉬운 머신러닝 문제

- 데이터의 변화가 천천히 일어난다. (분기, 월 단위)

- 모델 재학습이 다음에 의해 일어난다.
  - 더 많은 데이터로 모델 성능 개선
  - 소프트웨어 혹은 시스템의 변화

- 라벨링
  - 수집한 데이터
  - 크라우드소싱

머신러닝 문제의 특징

쉬운/어려운 머신러닝 문제

어려운 머신러닝 문제

- 데이터의 변화가 빠르게 일어난다. (주 단위)

- 모델 재학습이 다음에 의해 일어난다.
  - 모델 성능 저하
  - 더 많은 데이터로 모델 성능 개선
  - 소프트웨어 혹은 시스템의 변화

- 라벨링
  - 직접적인 피드백
  - 수집한 데이터
  - 크라우드소싱

머신러닝 문제의 특징

머신러닝 프로그래밍

작은 단위의 프로그래밍 (코딩)	큰 단위의 프로그래밍 (엔지니어링)
단일 코드	모듈식 설계 및 구현
재사용불가코드	재사용 가능한 라이브러리
문서화되지 않은 코드	문서화된 인터페이스 및 추상화
테스트되지 않은 코드	잘 테스트된 코드
코드 한 번으로 벤치마킹 되지 않거나 해킹	지속적으로 벤치마킹되고
최적화됨	최적화된 코드
검증되지 않은 코드	검토 및 피어 확인 코드
디버깅할 수 없는 코드 또는 특별 툴링	디버그 코드 및 디버그 툴링
계측되지 않은 코드	계측 가능 및 계측 코드

머신러닝 문제의 특징

머신러닝 프로그래밍 문제의 특징

작은 단위의 머신러닝 프로그래밍 (코딩)	큰 단위의 머신러닝 프로그래밍 (엔지니어링)
~~단일 코드~~ 고정된 데이터세트	변화하는 데이터셋과 metric
~~재사용불가코드~~ 통합 불가능한 아티팩트	재사용 가능한 모델들
~~문서화되지 않은 코드~~ 문제 정의 없음	문제 정의, 찾기 쉬운 아티팩트
~~테스트되지 않은 코드~~ 검증되지 않은 데이터셋, 모델	예상치, 데이터 검증, 모델 검증
코드 한 번으로 벤치마킹 되지 않거나 해킹 최적화됨	모델의 성능과 품질을 벤치마크
~~검증되지 않은 코드~~ 편향된 데이터셋, 아티팩트	[데이터, 모델] x [설명가능성, 공정성]
~~디버깅할 수 없는 코드 또는 특별 툴링~~	시각화, 요약, 이해
~~계측되지 않은 코드~~	아티팩트의 형상관리

머신러닝 문제의 특징

팀의 구성

팀의 구성:

ML 프로젝트는
Research Scientist, Research Engineer, Software Engineer로 구성됨

SW 프로젝트는 Software Engineer로 구성됨

머신러닝 문제의 특징

개발 프로세스

개발:

ML은 본질적으로 실험임
다른 Feature, 알고리즘, 모델링 기술 및 파라미터 구성을
시도해
가능한 빨리 문제점에 가장 적합한 것을 찾음

무엇이 효과가 있었는지, 무엇이 그렇지 않은지를 추적하고,
코드 재사용을 극대화하면서 재현성을 유지하는 것이
과제임

머신러닝 문제의 특징

테스팅 방법

테스팅:

ML 시스템 테스트는
SW 시스템 테스트보다 더
복잡함

머신러닝 문제의 특징

배포

테스팅:

ML 시스템에서,
배포는 오프라인에서 훈련된 ML 모델을 구축하는 것만큼
간단하지 않음

머신러닝 문제의 특징

프로덕션

프로덕션:

ML 모델은 코딩 뿐만 아니라
지속적으로 발전하는 데이터 프로파일 때문에 성능이 저하될 수 있다.
따라서 데이터의 통계치를 추적하고 모델의 온라인 성능을 모니터링해
예상치를 벗어날 때 알림을 보내거나 롤백해야 한다.

MLOps의 핵심 문제

트리거

진화하는 데이터셋

CI/CD/CT

모델 관리 시스템

MLOps의 핵심 문제

머신러닝 프로그래밍 문제의 특징

작은 단위의 머신러닝 프로그래밍 (코딩)	큰 단위의 머신러닝 프로그래밍 (엔지니어링)
~~단일 코드~~ 고정된 데이터세트	천화하는 데이터셋과 Metric
~~재사용불가코드~~ 통합 불가능한 아티팩트	재사용 가능한 모델들
~~문서화되지 않은 코드~~문제 정의 없음	문제 정의, 찾기 쉬운 아티팩트
~~테스트되지 않은 코드~~ 검증되지 않은	예상치, 데이터 검증, 모델 검증
데이터셋, 모델	모델의 성능과 품질을
코드 한 번으로 벤치마킹 되지 않거나 해킹	벤치마크
최적화됨	[데이터, 모델] x [설명가능성,
~~검증되지 않은 코드~~ 편향된 데이터셋,	공정성]
아티팩트	시각화, 요약, 이해
~~디버깅할 수 없는 코드 또는 특별 툴링~~	아티팩트의 형상관리
~~계측되지 않은 코드~~	

MLOps의 핵심 문제

모델 학습 / 배포 트리거

머신러닝 시스템은 다음과 같은 상황에서 학습과 모델을 배포한다

- 요청 시 : 파이프라인의 임시 수동 실행
- 일정 기준 : 라벨이 지정된 새 데이터는 매일, 매주 또는 매월
- 새 학습 데이터 : 새 데이터가 들어오는 경우 모델의 재학습을 트리거
- 모델 성능 저하 시 : 성능 저하가 눈에 띄는 경우 모델 재학습
- 데이터 분포의 중요한 변화 시(Concept Drift) :
  온라인 모델의 전체 성능을 평가하기는 어렵지만 예측을 수행하는 데
  사용되는
  피쳐의 데이터 분포에 큰 변화가 있으면, 모델이 오래되었다는 걸 뜻함

MLOps의 핵심 문제

머신러닝 프로그래밍 문제의 특징

진화하는 데이터셋과

Metric

MLOps의 핵심 문제

머신러닝 프로그래밍 문제의 특징

지속적 통합(Continuous Integration)

지속적 배포(Continuous Deployment)

지속적 학습(Continuous Training)

MLOps의 핵심 문제

머신러닝 프로그래밍 문제의 특징

지속적 학습(Continuous Training)

MLOps 성숙도 레벨

MLOps 성숙도 레벨 0

MLOps 성숙도 레벨 1

MLOps 성숙도 레벨 2

MLOps 성숙도 레벨 0

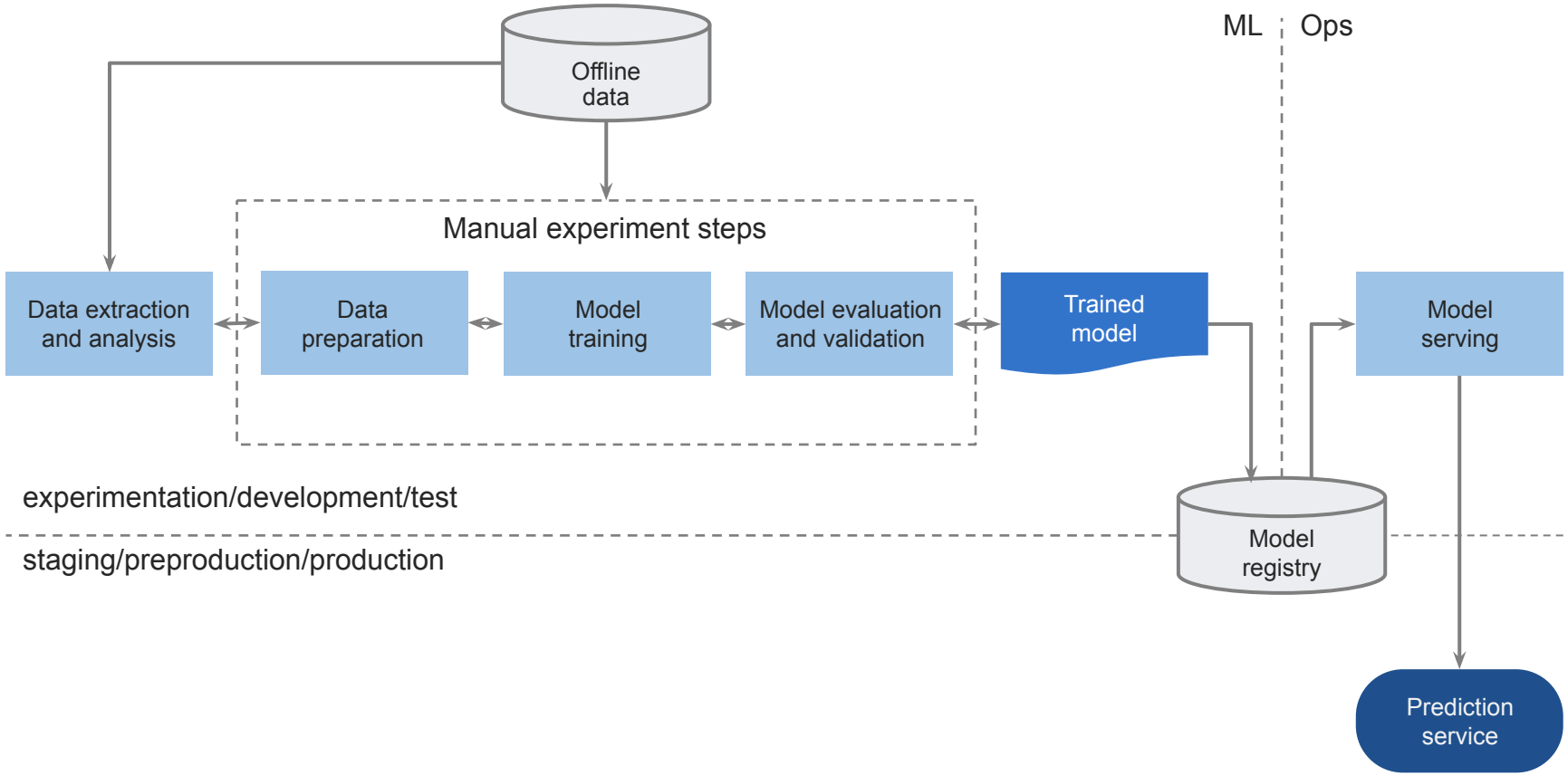
MLOps 성숙도 레벨 0의 특성

- 수동, 스크립트 중심, 대화식(Interactive) 프로세스
- ML과 운영의 분리
- 드문 릴리즈 반복
- CI 없음
- CD 없음
- 배포는 예측 서비스를 의미
- Active 성능 모니터링 부족

출처 : https://cloud.google.com/solutions/machine-learning/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

MLOps 성숙도 레벨 0

MLOps 성숙도 레벨 0의 특성



출처 : https://cloud.google.com/solutions/machine-learning/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

MLOps 성숙도 레벨 1

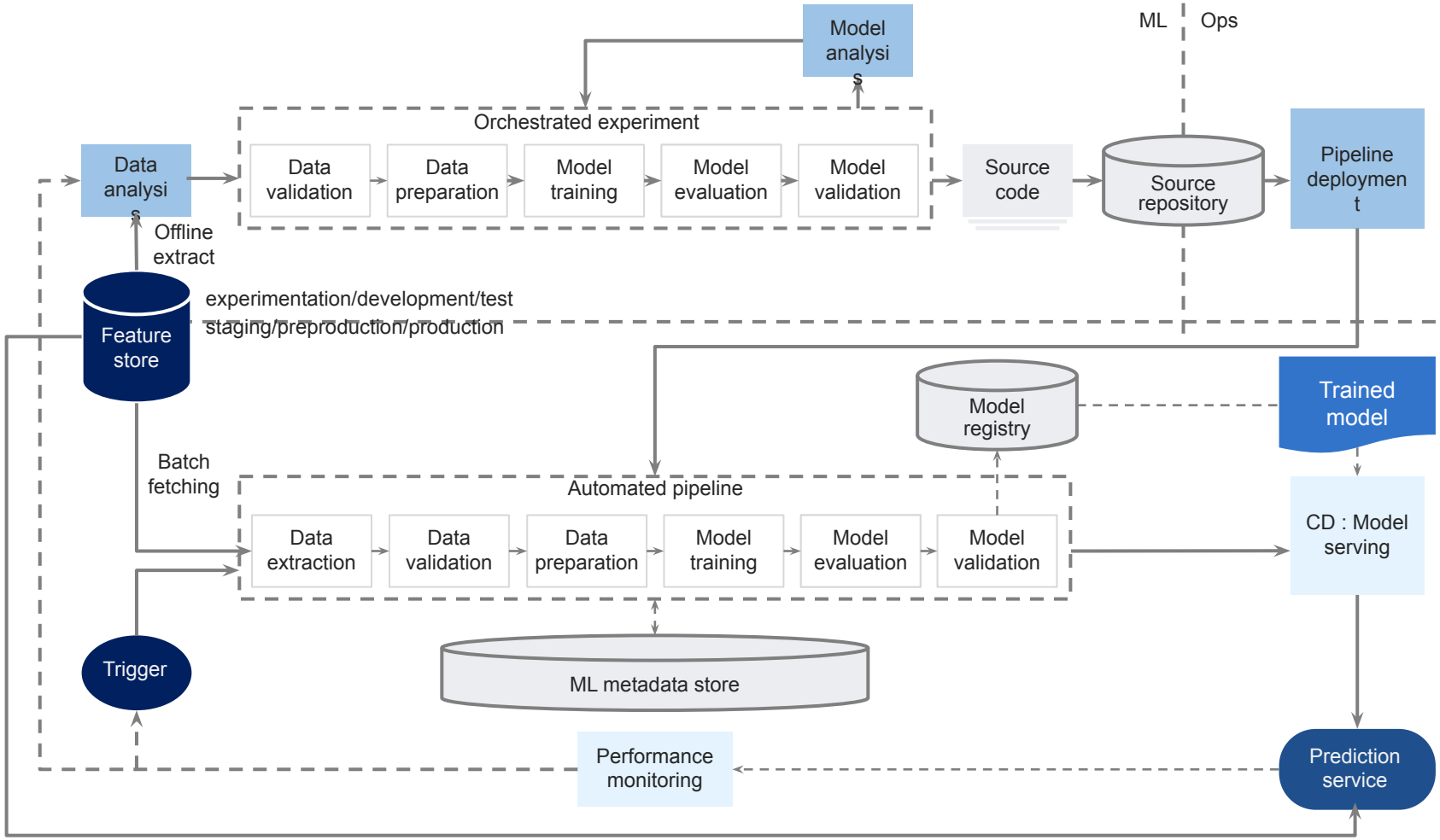
MLOps 성숙도 레벨 1의 특성

- 빠른 실험
- 프로덕션 모델의 CT
- 실험 운영 환경의 조화
- 구성 요소 및 파이프라인을 위한 모듈화된 코드
- 지속적인 모델 제공
- 파이프라인 배포

출처 : https://cloud.google.com/solutions/machine-learning/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

MLOps 성숙도 레벨 1

MLOps 성숙도 레벨 1의 특성



출처 : https://cloud.google.com/solutions/machine-learning/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

MLOps 성숙도 레벨 2

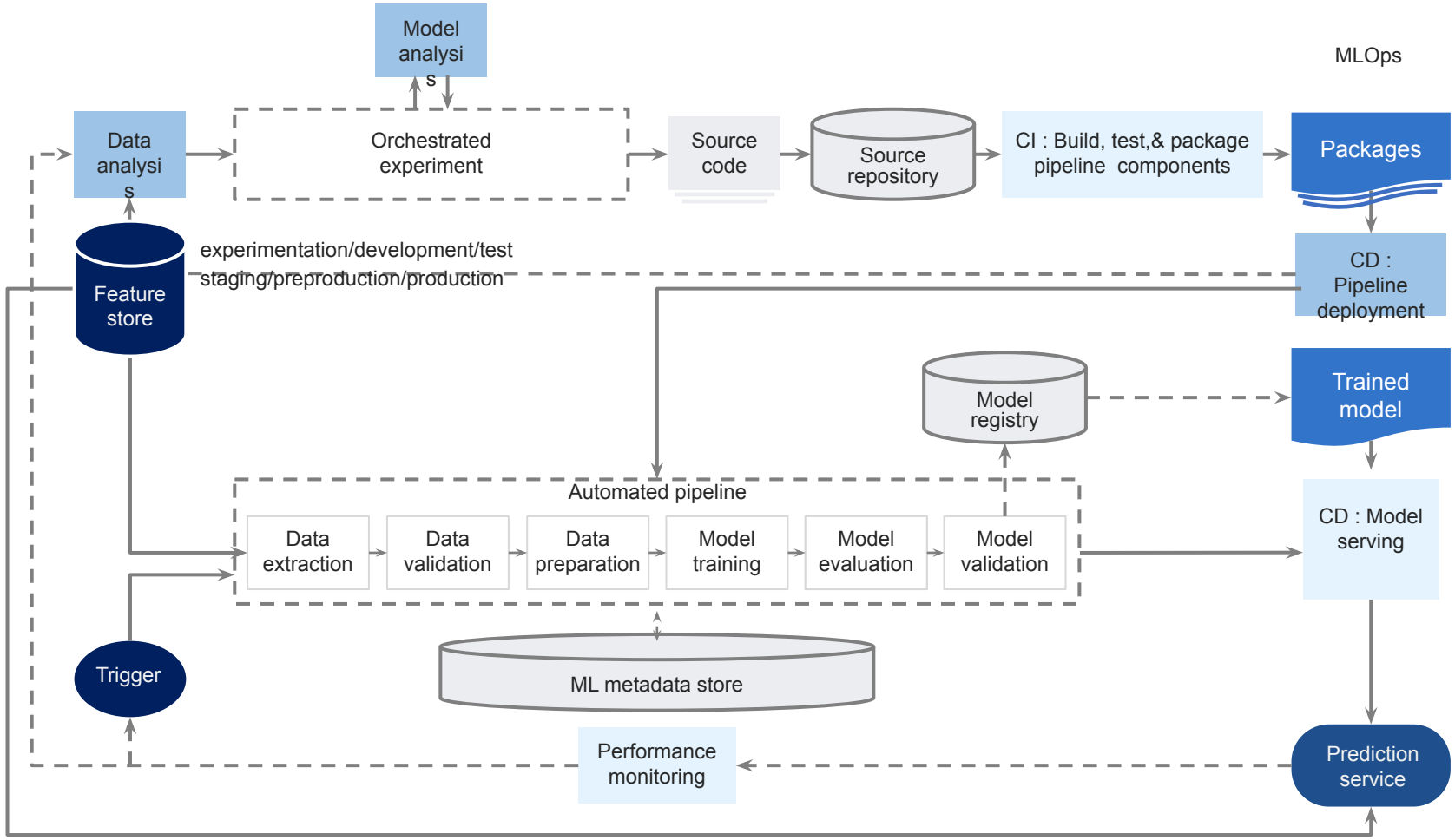
MLOps 성숙도 레벨 2의 특성

- Production에서 파이프라인을 빠르고 안정적으로 업데이트하려면 자동화된 CI/CD 시스템이 필요하다.
- 이 자동화된 CI/CD 시스템을 통해 데이터 과학자는 Feature Engineering, 모델 아키텍처 및 하이퍼파라미터에 대한 새로운 아이디어를 신속하게 탐색할 수 있다.

출처 : https://cloud.google.com/solutions/machine-learning/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

MLOps 성숙도 레벨 2

MLOps 성숙도 레벨 2의 특성



출처 : https://cloud.google.com/solutions/machine-learning/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

짚어보기

머신러닝 파이프라인의 이해

01. 머신러닝 파이프라인에 대해 이해한다.

머신러닝 파이프라인의 개념에 대해 이해한다.

02. 머신러닝 문제의 특징을 이해한다.

머신러닝 기반의 소프트웨어가 갖고 있는 특징을 이해한다.

03. MLOps의 핵심 문제에 대해 이해한다.

MLOps에서 풀어야 하는 핵심 문제에 대해 이해한다.

04. MLOps 성숙도 레벨에 대해 이해한다.

MLOps 성숙도 레벨에 따른 특징을 이해한다.

머신러닝 파이프라인

머신러닝 파이프라인의 이해

송호연

감사합니다.

THANKS FOR WATCHING