programmers

5. SageMaker 사용해보기

머신러닝 개발상의 난점
머신러닝 개발 프레임웍 필요성 대두
AWS SageMaker 소개

한기용

keeyonghan@hotmail.com

programmers

목차

programmers

4장 숙제 리뷰

보스턴 주택 가격 예측 모델을 Random Forest로 만들어 보자

숙제

Regression 모델을 Random Forest로 만들어보기

- 구글 Colab 보며 리뷰하기

```
from sklearn.ensemble import RandomForestRegressor
…
rf = RandomForestRegressor(
    n_estimators=100,   # 트리의 갯수로 디폴트 값도 100
    random_state=42
)
rf.fit(X_train, y_train)
```

programmers

머신러닝 모델 개발 절차

머신러닝 모델 개발을 구성하는 절차들을 처음부터 끝까지 살펴보자

모델 개발 과정

1. 문제 정의: 모델 개발 당위성을 가설로 제시
2. 데이터 수집 및 분석: 훈련용 데이터
3. 모델 훈련 및 테스트
4. **모델 배포**
5. 모델 성능 A/B 테스트
6. 전체 배포 여부 결정

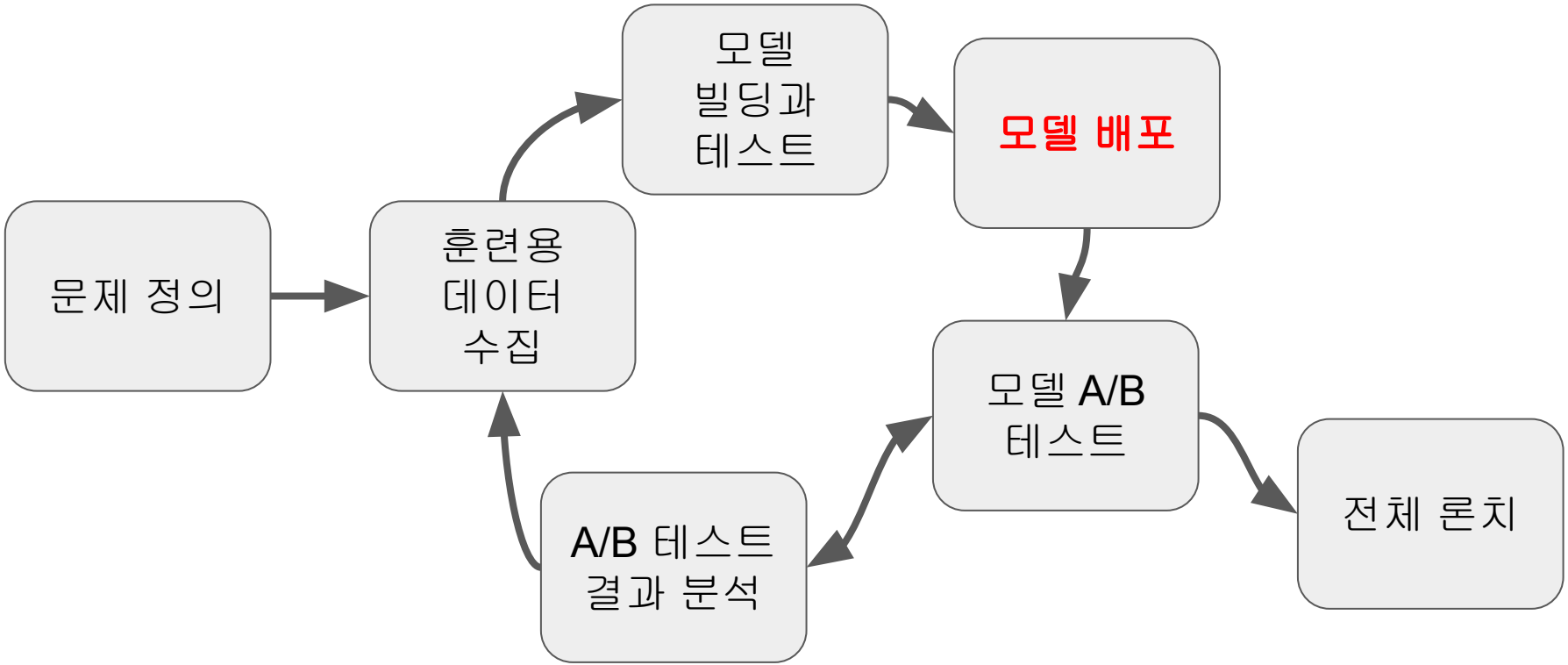
모델 개발 과정 - 가설

- 어떤 문제를 해결하려고 하며 왜 머신러닝이 필요한가?
- 문제 해결의 성공 여부를 결정하는 지표
  - 지표는 어떻게 계산되며 성공과 실패의 기준은 무엇인가?
- 가설을 통해 풀려고 하는 문제의 임팩트와 중요도를 가늠할 수 있음

모델 성능 A/B 테스트

- 훈련용 데이터를 가지고 한 검증만으로 불충분
- 만든 모델을 실제 사용자의 반응을 통해 어떻게 검증할 수 있나?
  - 기존 방식과 얼마나 다른지 AB 테스트를 사용해 검증

모델 개발 전체 과정 (Life-Cycle)



잠깐! A/B 테스트란? (1)

- 온라인 서비스에서 새 기능의 임팩트를 객관적으로 측정하는 방법
  - 의료쪽에서 무작위 대조 시험(Randomized Controlled Trial)
- 새로운 기능을 론치함으로 생기는 위험부담을 줄이는 방법
  - 100%의 사용자에게 론치하는 것이 아니라 작게 시작하고 관찰 후 결정
  - 실제 예제: 추천을 기계학습기반으로 바꾼 경우
    - 먼저 5%의 사용자에게만 론치하고 나머지 95%의 사용자와 매출액과 같은 중요 지표를 가지고 비교
    - 5% 대상으로 별문제 없으면 10%, 20% 이런 식으로 점진적으로 키우고 최종적으로 100%로 론치

잠깐! A/B 테스트란? (2)

- 보통 사용자들을 2개의 그룹으로 나누고 시간을 두고 관련 지표를 비교
  - 한 그룹은 기존 기능에 그대로 노출 (control)
  - 다른 그룹은 새로운 기능에 노출 (test)
- 가설에서 영향받는 지표를 미리 정하고 시작하는 것이 일반적
  - 지표의 경우 성공/실패 기준까지 생각해보는 것이 필요

“If you can’t measure it, you can’t improve it”

William Thomson, Lord Kelvin

programmers

머신러닝 모델 개발시 자주 발생하는 문제점들

가장 큰 문제는 배포와 모델을 다시 빌딩하는 순간에 발생한다

어떤 문제점들이 존재하는가?

- 훈련용 데이터 셋 관리
- ML 모델 빌딩시 사용 하이퍼 파라미터 관리
- ML 모델 관리
- 모델 론치 프로세스
- 모델 AB 테스트 프로세스

훈련용 데이터 셋 관리

- 어떻게 훈련용 데이터 셋을 수집했는가?
  - label 데이터 비율?
  - bias가 있는가?
- 어떻게 이 데이터들을 보관하고 관리할 것인지?
- 다양한 feature들을 어떻게 구현했고 관리할 것인가?
  - 데이터 과학자가 코딩을 할 수 있는지 여부가 아주 중요해짐
    - 유닛 테스트를 통한 버그 줄이기. 모델 개발 속도 단축
    - 야후 예
  - Feature Store가 필요해짐
    - 이미 만들어진 feature들의 공유와 재사용성이 중요해짐

ML 모델 빌딩과 검증

- 다양한 러닝 알고리즘과 하이퍼 파라미터를 어떻게 손쉽게 테스트 가능한가?
  - 자동화가 관건
- 모델 빌딩에 사용한 하이퍼 파라미터를 저장하고 쉽게 찾을 수 있는가?
- 모델 검증 결과를 저장하고 쉽게 찾을 수 있는가?

ML 모델 관리

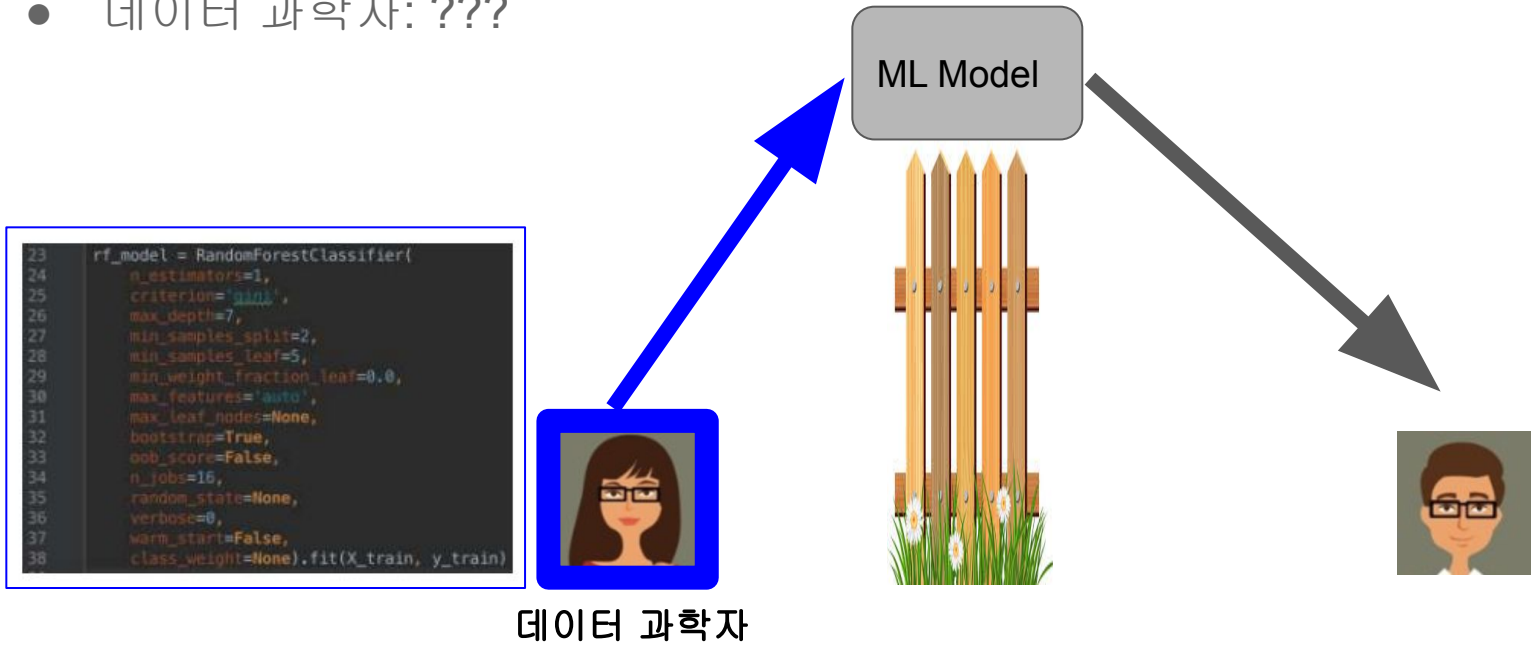
- 모델별 사용 알고리즘/하이퍼파라미터/트레이닝셋/테스트 결과를 유기적으로 보관해야함
  - 버전닝이 필요해짐
- 모델 재연성 (reproducibility)이 아주 중요해짐

ML 모델 론치 프로세스 (1)

- 만든 모델을 어떻게 프로덕션으로 론치할 것인가?
- 프로덕션 엔지니어링 팀과의 협업이 중요해짐
- 이 부분 프로세스가 자동화가 필요

모델 개발시 데이터 과학자들의 일반적인 생각

- 데이터 과학자: 아주 좋은 머신러닝 모델을 만들고 말겠어!
- 엔지니어: 만든 모델 어떻게 론치하지?
- 데이터 과학자: ???

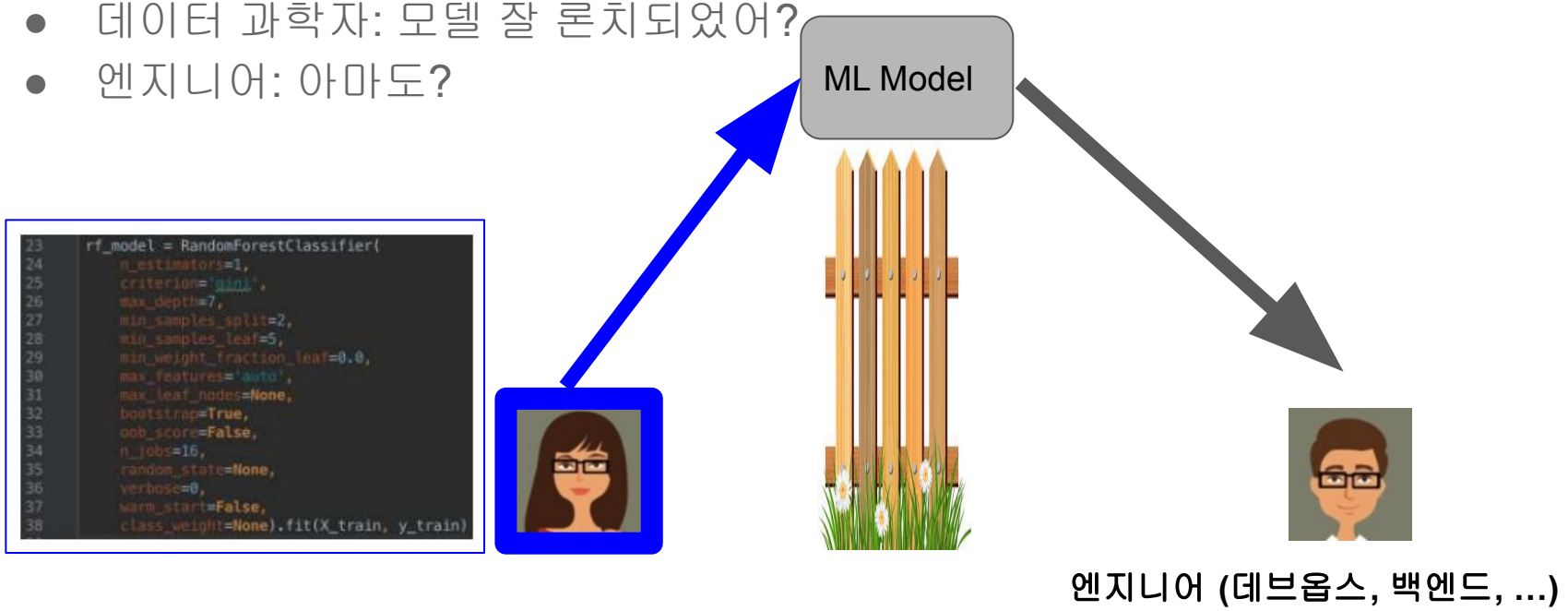


모델 개발시 엔지니어들의 일반적인 생각

- 엔지니어: 머신러닝 모델을 론치하는 것 귀찮네. 아몰랑

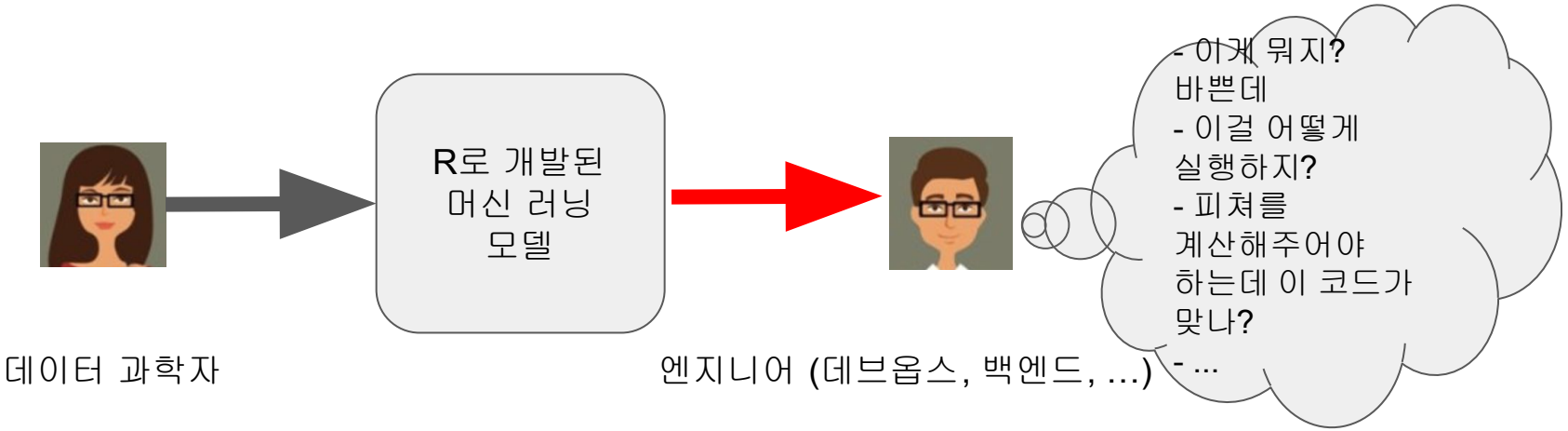
(시간이 지난 후)

- 데이터 과학자: 모델 잘 론치되었어?
- 엔지니어: 아마도?



마찰이 생기는 지점 - 개발된 모델의 이양 관련

- 많은 수의 데이터 과학자들은 R을 비롯한 다양한 툴로 모델 개발
- 하지만 실제 프로덕션 환경은 다양한 모델들을 지원하지 못함
  - 개발/검증된 모델의 프로덕션 환경 론치시 시간이 걸리고 오류 가능성이 존재
  - 심한 경우 모델 관련 개발을 다시 해야함 (피쳐 계산과 모델 실행 관련)



programmers

머신러닝 모델 개발 프레임웍의 필요성

앞서 문제를 해결하기 위한 프레임웍들이 필요해지기 시작

앞서 모든 과정을 처음부터 끝까지 해주는 툴이 있다면?

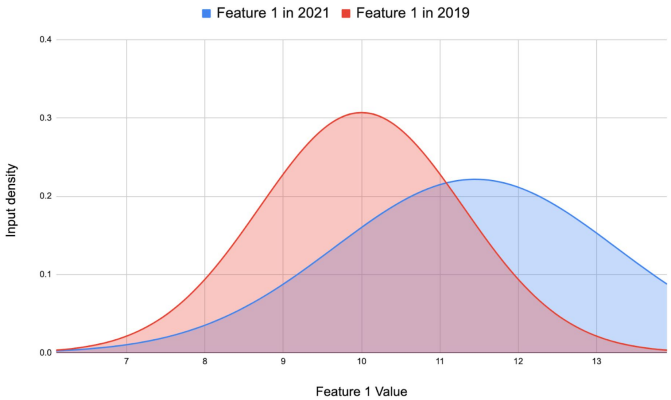
- 다양한 회사에서 자체 프레임웍을 개발
  - 우버의 미켈란젤로
  - 에어비앤비의 빅헤드 (BigHead)
  - 넷플릭스의 메타플로우 (Metaflow)
  - 리프트의 플라이트 (Flyte)
- 클라우드 업체들도 프레임웍을 SaaS 형태로 제공
  - AWS의 SageMaker
  - Google Cloud의 TFX, Kubeflow, and AI Platform
  - Azure의 Machine Learning

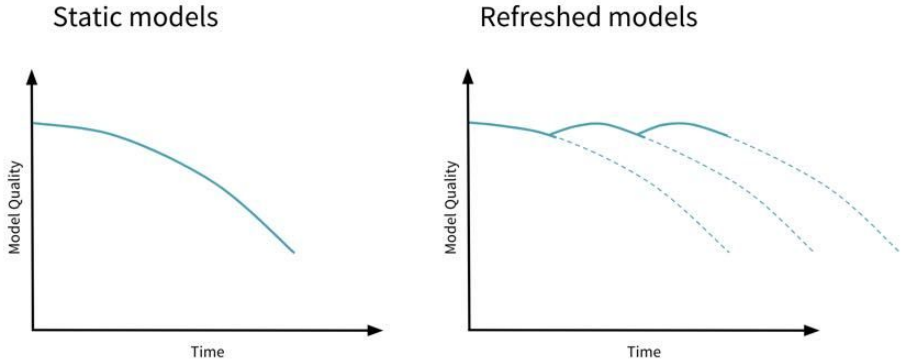
MLOps 직군의 도래

- 개발자 직군에서 DevOps에 해당
  - DevOps가 하는 일은
    - 개발자가 만든 코드를 시스템에 반영하는 프로세스 (CI/CD, deployment)
    - 시스템이 제대로 동작하는지 모니터링 그리고 이슈 감지시 escalation 프로세스
      - On-call 프로세스
- MLOps가 하는 일은?
  - 앞의 DevOps가 하는 일과 동일. 차이점은 개발자 코드가 아니라 ML 모델이 대상이 된다는 점
  - 어떻게 ML모델 개발을 더 쉽게 하고 자동화하여 자주 모델 빌딩을 하고 이를 프로덕션으로 론치할 수 있는가? 또한 모델 서빙 환경에 문제가 생기나 모델의 성능이 떨어질 경우 이를 어떻게 감지하고 조치를 취할 수 있는가?

Data Drift로 인한 모델 성능 저하

- ML 모델에서 가장 중요한 것은 훈련 데이터
- 시간이 지나면서 훈련에 사용한 데이터와 실제 환경의 데이터가 다르게 변화함
  - 이를 Data drift라고 부르며 이를 모니터링하는 것이 중요
- 즉 주기적으로 ML 모델을 다시 빌딩해주는 일이 필요





MLOps vs. DevOps

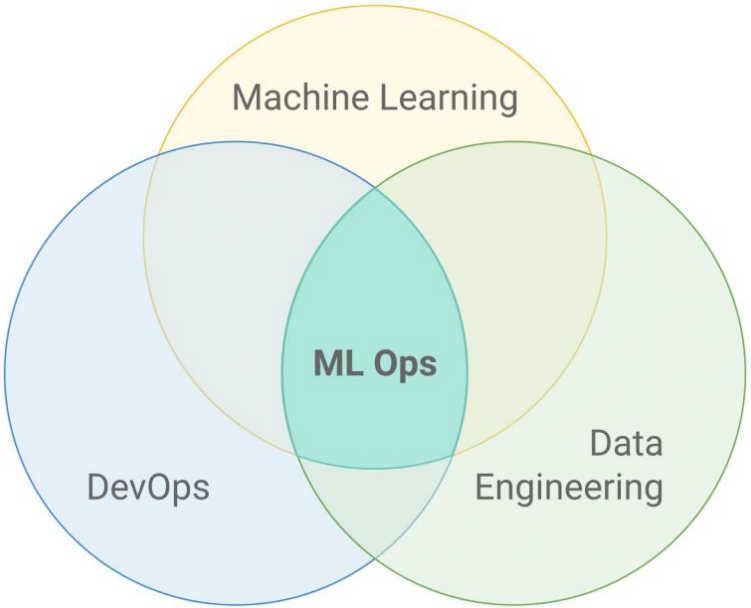
- DevOps가 하는 일은?
  - Deliver software faster and more reliably in automated fashion
  - 개발자가 만든 코드를 시스템에 반영하는 프로세스 (CI/CD)
  - 시스템이 제대로 동작하는지 모니터링 그리고 이슈 감지시 escalation 프로세스 수행
    - On-call 프로세스
- MLOps가 하는 일은?
  - Deliver ML models faster and more reliably in automated fashion
  - 앞의 DevOps가 하는 일과 동일. 차이점은 개발자 코드가 아니라 ML 모델이 대상이 된다는 점
  - 모델을 계속적으로 빌딩하고 배포하고 성능을 모니터링
    - ML모델 빌딩과 프로덕션 배포를 자동화할 수 있을까? 계속적인 모델 빌딩(CT)과 배포!
  - 모델 서빙 환경과 모델의 성능 저하를 모니터링하고 필요시 escalation 프로세스 진행
    - Latency의 중요성
    - Data drift 측정

CI & CD

- CI (Continuous Integration)
  - Developers frequently merge code changes into a central repo
  - Building and testing are automated
- CD (Continuous Delivery or Deployment)
  - Passing builds (packages) are deployed directly to the production environment

MLOps 엔지니어가 알아야하는 기술

- 데이터 엔지니어가 알아야 하는 기술
  - 파이썬/스칼라/자바
  - 데이터 파이프라인과 데이터 웨어하우스
- DevOps 엔지니어가 알아야 하는 기술
  - CI/CD, 서비스 모니터링, …
  - 컨테이너 기술 (K8S, 도커)
  - 클라우드 (AWS, GCP, Azure)
  - Infrastructure As Code (Configuration As Code)
- 머신러닝 관련 경험/지식
  - 머신러닝 모델 빌딩과 배포
  - ML 모델 빌딩 프레임웍 경험
    - SageMaker, Kubeflow, MLflow



https://builtin.com/machine-learning/mlops

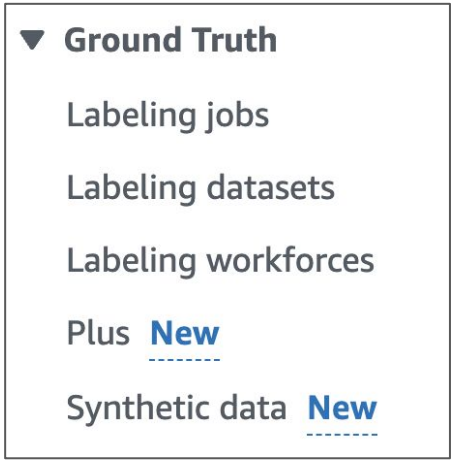
programmers

SageMaker 소개

AWS의 ML end-to-end Framework인
SageMaker를 사용해보자

Amazon SageMaker란? (1)

- 머신러닝 모델 개발을 처음부터 끝까지 해결해주는 AWS 서비스
- 크게 4가지 기능 제공
  - 트레이닝 셋 준비 (Ground Truth)
  - 모델 훈련
  - 모델 검증
  - 모델 배포와 관리: API 엔드포인트, 배치 서빙, …
- 다양한 머신러닝 프레임웍을 지원
  - Tensorflow/Keras, PyTorch, MXNet, …
  - 자체 SageMaker 모듈로 머신러닝 모델 훈련 가능

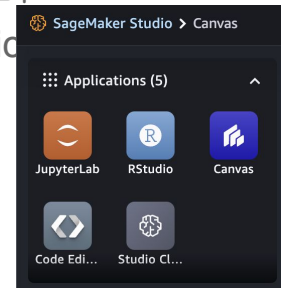


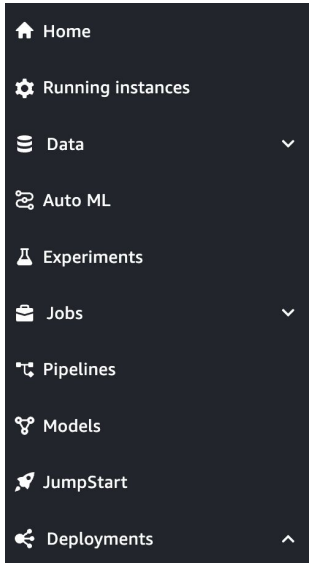
Amazon SageMaker란? (2)

- 다양한 개발방식 지원
  - 기본적으로 Python Notebook (SageMaker 모듈)을 통해 모델 훈련
    - 스칼라/자바 SDK도 제공
  - AutoPilot이라는 코딩 불필요 모델 훈련 기능 제공
    - 이 경우에도 코드를 만들어줌 (“Download Notebook”)
    - 새 버전 (2024.01)에서는 SageMaker Canvas 밑으로 기능이 이전했음
- 다른 클라우드 업체들도 비슷한 프레임웍 제공

Amazon SageMaker 기능: SageMaker Studio

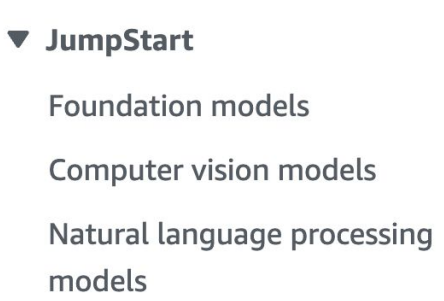
- 이전 Studio는 주피터 노트북이었는데 이제는 Studio Classic이라 부름
- 이제는 다수의 프로그램을 호스팅하는 환경으로 변화
  - 주피터 노트북, RStudio, Canvas, Code Editor, Studio Classic
- 아래와 같은 기능 제공
  - Machine Learning IDE라고 자칭
  - Data Wrangler: Processing, Data Sources, Feature Store
  - Studio Notebooks: Algorithms, Autopilot, JumpStart
  - One-Click Training: Experiments, Automatic Model Tuning, Debugger
  - One-Click Deployment: Multi-model Endpoints, Model Monitor, Pipelines

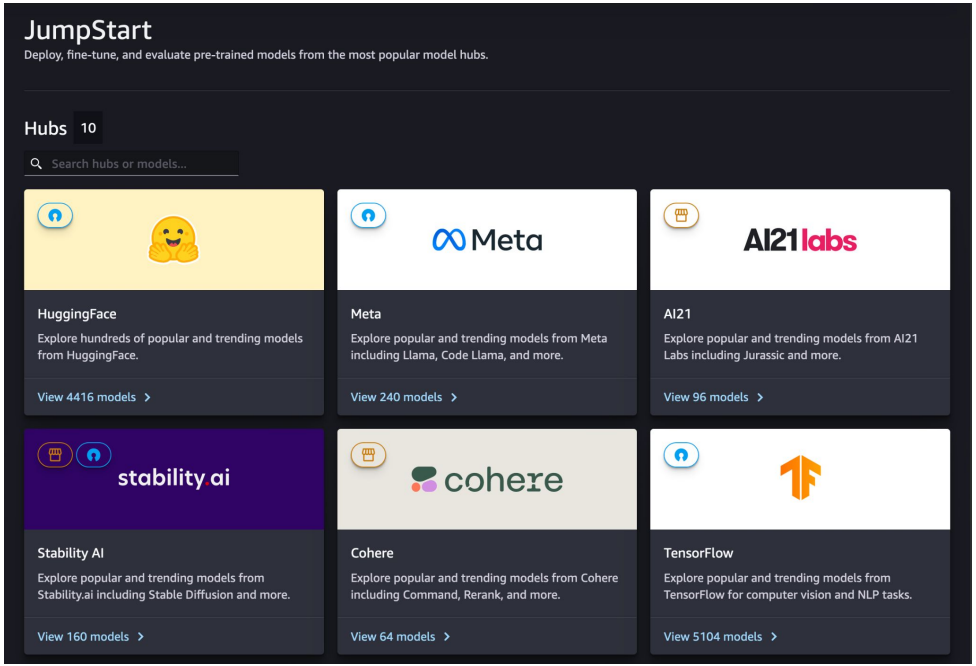




Amazon SageMaker 기능: SageMaker Jumpstart

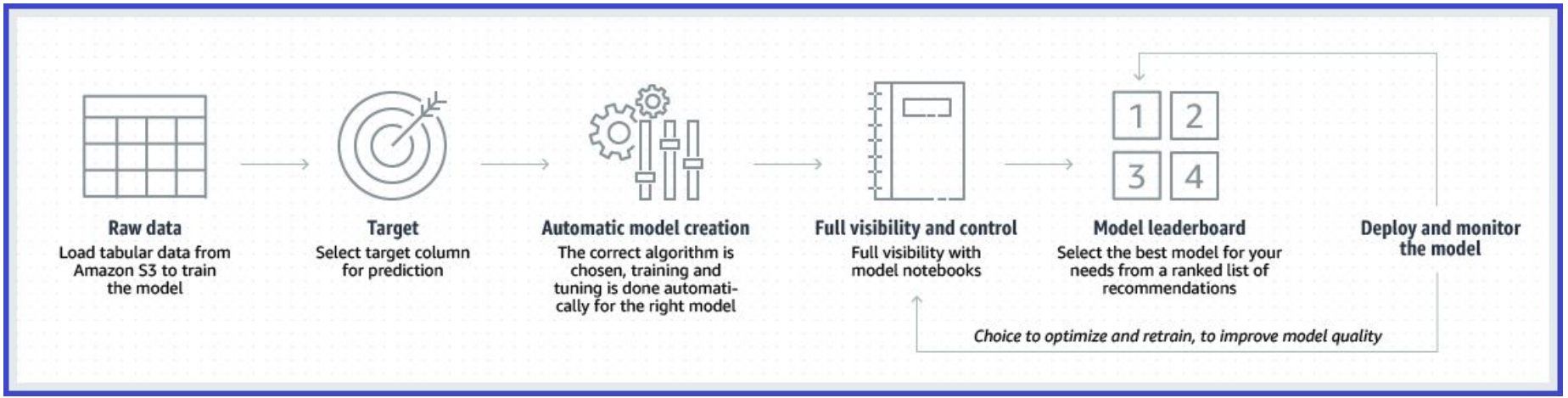
- 다양한 범위의 문제를 해결할 수 있는 pre-trained 오픈소스 모델 제공
- Hugging Face Models나 Kaggle의 Models와 흡사





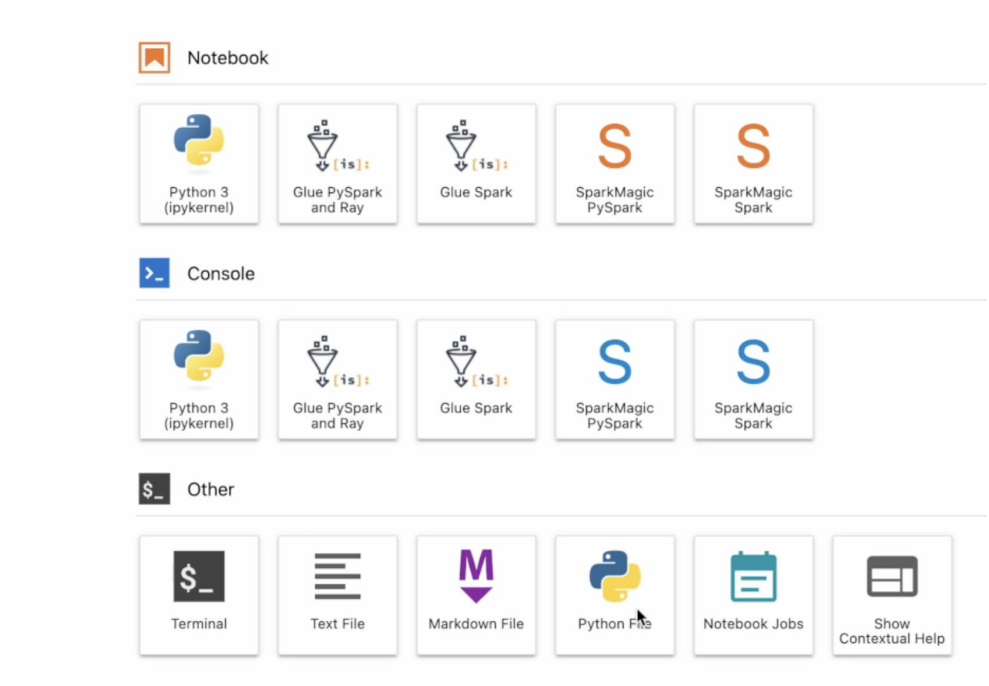
Amazon SageMaker 기능: Canvas (Low/No code)

- Data analysis and preprocessing
- Model selection
- Hyperparameter optimization
- Model training and evaluation
- Model deployment (and monitor)



AWS SageMaker Studio의 주피터랩의 실행화면

- 자체 SageMaker 파이썬 모듈을 가지고 실행됨 (서버 사양 선택 필요)



xgboost 학습 알고리즘이란?

- e**X**treme **G**radient **B**oosting
- 앙상블 기반 Decision Tree를 Gradient Boosting으로 분산환경에서 구현한 알고리즘
- 분류와 회귀 문제 모두에 적합
- 속도와 성능 측면에서 좋은 결과를 보여줌
- SageMaker에서 가장 효율적인 학습 알고리즘

programmers

SageMaker Autopilot 실습

SageMaker가 제공해주는 Autopilot에 대해 배우고
이어서 실습을 해보자

SageMaker Canvas의 AutoML 소개

- AutoPilot: SageMaker Canvas에서 제공되는 AutoML 기능
  - AutoML이란 모델빌딩을 위한 훈련용 데이터 셋을 제공하면 자동으로 모델을 만들어주는 기능
- AutoPilot은 훈련용 데이터 셋을 입력으로 다음을 자동으로 수행
  - 먼저 데이터 분석(EDA: Exploratory Data Analysis)을 수행
  - 두 가지의 빌드 옵션: Standard와 Quick
    - Standard 옵션: 오래 걸리고 돈이 더 들지만 품질이 더 좋음
      - 다수의 머신 러닝 알고리즘과 하이퍼 파라미터의 조합에 대해 아래 작업을 수행
        - 머신 러닝 모델을 만들고 훈련하고 테스트하고 테스트 결과를 기록
    - Quick 옵션: 10-15분 정도로 빠르게 모델을 하나 만들어줌. 비용/시간과 성능 간의 trade-off
  - 파이썬 노트북 코드를 나중에 다운로드해서 직접 개선 후 사용하는 것도 가능
    - 즉 AutoPilot 기능을 통해 모델개발 속도를 단축하는 것이 가능

AutoPilot - 당뇨병 환자의 재입원 여부 예측

- 튜토리얼 참고
- 데모 진행
  - AWS에 익숙한 사람이 아니라면 그냥 데모만 보는 걸 추천
  - 그 이유는 사용된 리소스 삭제가 안되면 돈이 줄줄 샐 가능성이 있기 때문인데 SageMaker domain 삭제가 생각보다 쉽지 않음.

programmers

SageMaker 마무리

오늘 배운 것을 정리해보도록 하자

학습 평가

번호	문제	보기
1	다음 중 일반적인 모델 개발 절차 중의 하나가 아닌 것은?	1) 문제 정의 2) 훈련용 데이터 셋 수집 3) 새로운 머신러닝 알고리즘 개발 4) 모델 배포
2	다음 중 일반적인 모델 개발시 발생하는 문제가 아닌 것은?	1) 훈련용 데이터 셋 관리 2) 모델 빌딩시 사용한 하이퍼 파라미터 관리 3) 모델 배포시 시간 지연 4) 데이터 웨어하우스의 용량 부족
3	다음 중 SageMaker의 특징이 아닌 것은?	1) 머신러닝 모델 개발과 관련된 작업을 한 곳에서 가능 2) AutoPilot을 통해 훈련용 데이터 셋을 입력으로 최적의 모델을 자동생성 가능 3) 훈련용 데이터 셋을 자동으로 수집해줌

학습 평가

번호	문제	보기	정답	해설
1	다음 중 일반적인 모델 개발 절차 중의 하나가 아닌 것은?	1) 문제 정의 2) 훈련용 데이터 셋 수집 3) 새로운 머신러닝 알고리즘 개발 4) 모델 배포	3)	새로운 머신러닝 알고리즘 개발은 일반적인 모델 개발 절차가 아님
2	다음 중 일반적인 모델 개발시 발생하는 문제가 아닌 것은?	1) 훈련용 데이터 셋 관리 2) 모델 빌딩시 사용한 하이퍼 파라미터 관리 3) 모델 배포시 시간 지연 4) 데이터 웨어하우스의 용량 부족	4)	데이터 웨어하우스의 용량 부족은 머신러닝 모델 개발과는 직접적인 관련이 없음
3	다음 중 SageMaker의 특징이 아닌 것은?	1) 머신러닝 모델 개발과 관련된 작업을 한 곳에서 가능 2) AutoPilot을 통해 훈련용 데이터 셋을 입력으로 최적의 모델을 자동생성 가능 3) 훈련용 데이터 셋을 자동으로 수집해줌	3)	SageMaker Ground Truth라고 해서 훈련용 데이터셋과 관련된 것이 있으나 이는 데이터 수집을

머신러닝 프레임웍의 미래: 채팅 연동

- SageMaker의 경우 강력한 기능이 많지만 사용법이 쉽지 않음
- 어느 정도는 ChatGPT와 같이 채팅으로 사용하는 형태로 바뀌지 않을까 싶음
  - "내가 지금 만든 모든 모델 다 리스트해줘"
  - "모델 A의 최신버전을 배포해줘"
  - "모델 A의 최신버전의 성능을 어떤 metric로 평가했고 뭐였지?"
  - "모델 A가 버전이 몇개나 있지"
  - "모델 A의 API endpoint가 몇개가 있지?"

programmers

Q & A

오늘 강의에 대해서 궁금한 부분이 있으면 알려주세요!