

AWS Sagemaker를 이용한 MLOps 와 LLMOps

서진호 (synabreu@outlook.com)

<https://github.com/synabreu/awskrusergroup>

간단한 저의 소개



- Microsoft, Softleaf, .NET Soft, Ahnlab 등 약 25년 이상 간의 IT 경력 (Web, Mobile, Cloud, AI/ML)
- 서울과학종합대학원 AIMBA + FUS 석사과정
- 저서
 - AWS 기반 데이터 과학, 공동 번역(2023)
 - 마이크로소프트 전략과 미래 (2005)
- 블로그
 - <https://www.brunch.co.kr/@synabreu>
 - <https://github.com/synabreu/awskrusergroup>

어떻게 책을 적게 되었는가?

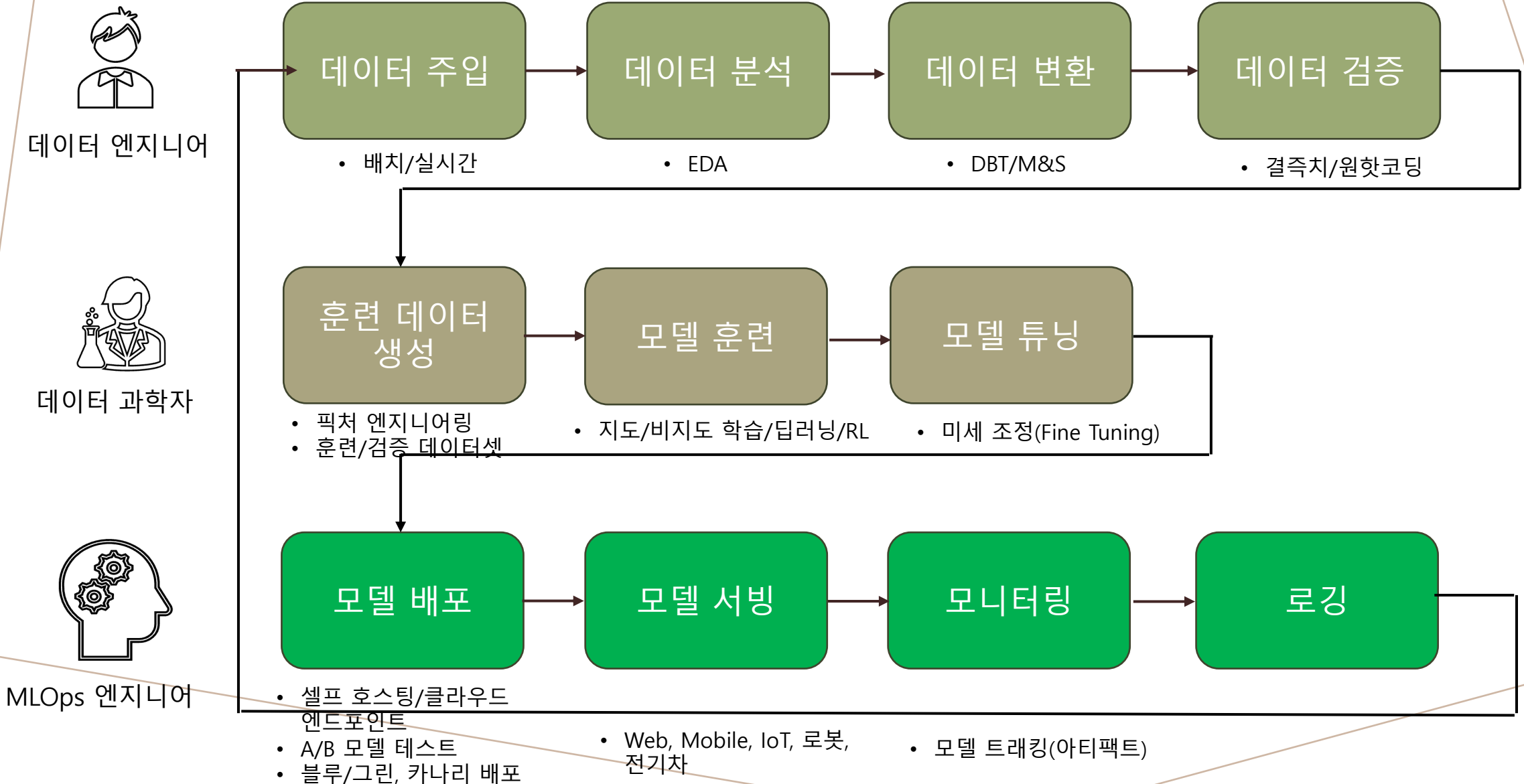


- 코로나 기간 동안 기업 영업을 하기 위한 AI/ML 컨설팅 교재 필요성
- 국내 퍼블릭 최대 클라우드 점유율
- 국내 개발자, 데이터 과학자, ML 엔지니어
- 책의 장점
 - AWS 데이터 과학과 AI/ML AWS SageMaker 총망라
- 책의 단점
 - 초급자의 자습서 형식이 아닌 중급자의 AI/ML 및 AWS Sagemaker 용어와 개념 설명, 코드 조각 위주

오늘의 첫번째 이야기

- MLOps (AI/ML lifecycle) 및 AI/ML Stack
- AWS SageMaker 2022.12 주요 기능
 - SageMaker Domain 및 User Profile
 - SageMaker Projects – Pipeline
 - SageMaker Notebooks
 - SageMaker Studio
 - SageMaker Processing (전처리)
 - SageMaker Autopilot (AutoML)
 - SageMaker Training Jobs
 - SageMaker Experiments 및 Trials
 - SageMaker Serving and Deployment
 - SageMaker Endpoints
 - SageMaker Debugger
 - SageMaker Monitor
- 실습: SageMaker Notebooks 에서 ML 모델을 ECR 배포 및 엔드포인트

전체 인공지능 개발 사이클 (MLOps)

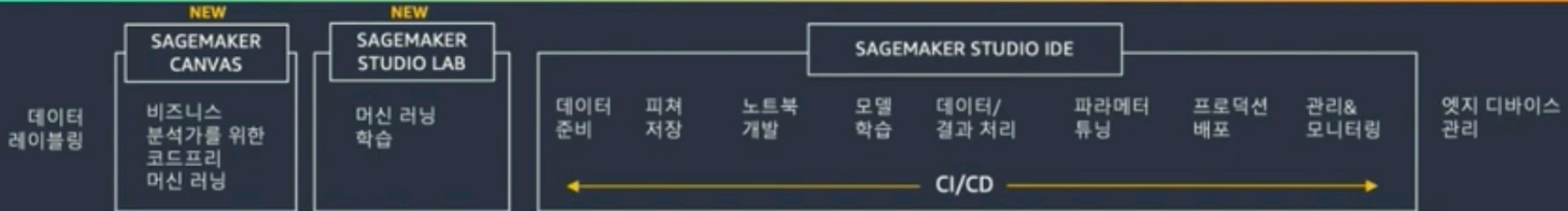


AI/ML Stack

AI 서비스



AMAZON SAGEMAKER



ML 프레임워크 & 인프라



SageMaker Domain 과 User Profile



- Amazon SageMaker 도메인: 1) Amazon EFS, 승인된 사용자 목록; 다양한 보안, 앱 정책 및 Amazon VPC 구성하고, 2) Amazon SageMaker Studio, Amazon SageMaker Studio Notebooks, SageMaker Canvas 및 RStudio를 사용하려면 Amazon SageMaker 도메인 온보딩 프로세스를 완료해야 한다. (SageMaker 서비스에서 Domain 클릭)
- *UserProfile* : 도메인 내의 단일 사용자를 나타냄. 공유, 보고 및 다른 사용자 중심 기능을 제공하고 도메인 온보딩 프로세스 때 생성됨.
- *Shared Spaces* : 공유한 JupyterServer 앱과 디렉토리로 구성됨. 도메인 내의 모든 사용자는 Shared Spaces에 액세스할 수 있고, 도메인의 모든 사용자 프로파일은 도메인의 모든 Shared Spaces에 액세스할 수 있음.
- *App* : 사용자의 노트북, 터미널, 콘솔의 읽기 및 실행 경험을 지원하는 앱이 나타냄. 앱 유형은 JupyterServer, KernelGateway, RStudioServerPro 또는 RSession일 수 있음. user는 여러 앱을 동시에 활성화할 수 있음.

Amazon SageMaker

PREPARE →

SageMaker Ground Truth

Label training data for machine learning

SageMaker Data Wrangler

Aggregate and prepare data for machine learning

SageMaker Processing

Built-in Python, Built-in Spark, BYO

SageMaker Feature Store

Store, update, retrieve, and share features

SageMaker Clarify

Detect bias and understand model predictions

BUILD →

SageMaker Studio Notebooks

Jupyter notebooks with elastic compute and sharing

Built-in and Bring-your-own Algorithms

Dozens of optimized algorithms or bring your own

Local Mode

Test and prototype on your local machine

SageMaker Autopilot

Automatically create machine learning models with full visibility

SageMaker JumpStart

Pre-built solutions for common use cases

TRAIN & TUNE →

One-click Training

Distributed infrastructure management

SageMaker Experiments

Capture, organize, and compare every step

Automatic Model Tuning

Hyperparameter optimization

Distributed Training

Training for large datasets and models

SageMaker Debugger

Debug and profile training runs

Managed Spot Training

Reduce training cost by 90%

DEPLOY & MANAGE →

One-click Deployment

Fully managed, ultra low latency, high throughput

Kubernetes & Kubeflow Integration

Simplify Kubernetes-based machine learning

Multi-model Endpoints

Reduce cost by hosting multiple models per instance

SageMaker Model Monitor

Maintain accuracy of deployed models

SageMaker Edge Manager

Manage and monitor models on edge devices

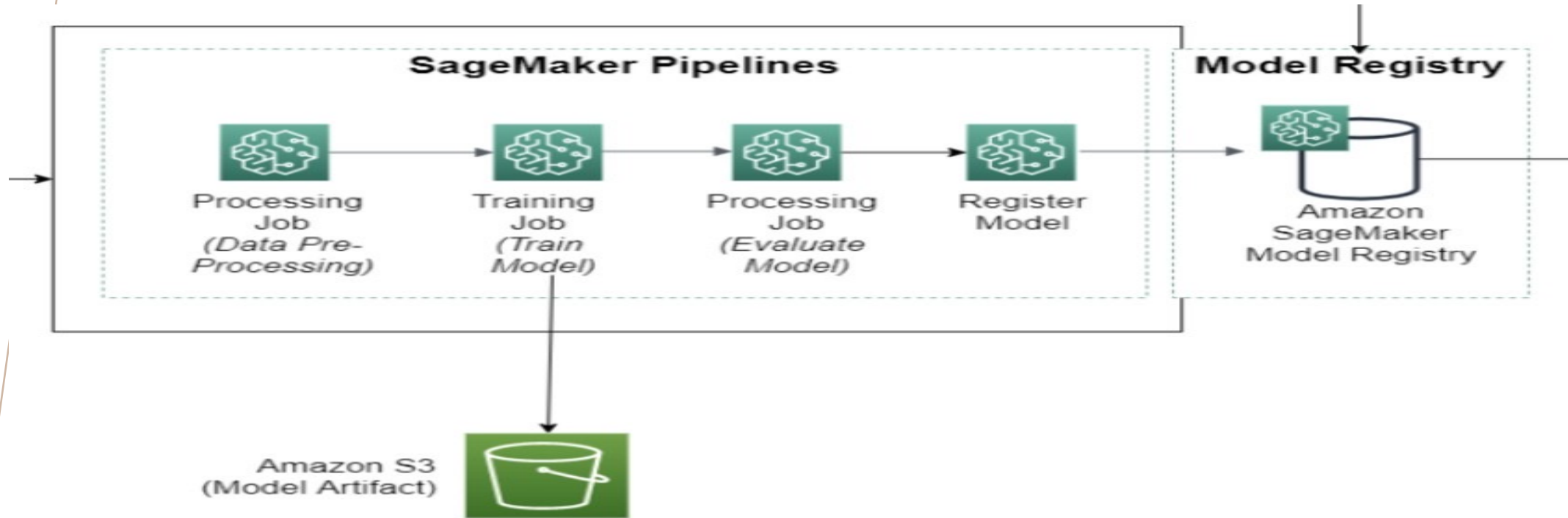
SageMaker Pipelines

Workflow orchestration and automation

SageMaker Studio

Integrated development environment (IDE) for ML

SageMaker Projects



- SageMaker Studio의 네이티브 MLOps 솔루션(CI/CD)
 - 도커 이미지 빌드
 - 데이터 전처리, 피처 엔지니어링
 - 모델 훈련, 모델 평가, 모델 배포
 - 모델 모니터링 및 신규 업데이트

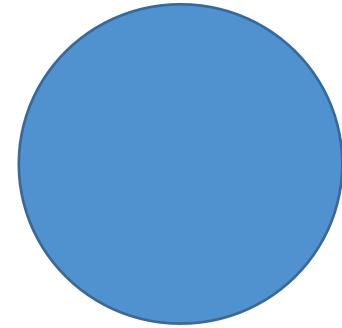
- ML 솔루션 빌드 및 배포용 코드 레포지토리 사용
- 세이지메이커 파이프라인으로 단계 정의

AI/ML 인스턴스 타입 선택하기

- 운영 보다는 “모델링” 중점
- 딥러닝에 의존하는 알고리즘은 훈련용 GPU 인스턴스(P3, g4dn)가 이익임.
 - P3 이나 g4dn 을 처음 사용하는 분은 AWS Support 에 Quota 신청 (2-3일 걸림)
- 추론은 컴퓨트 인스턴스(C5) 와 요청이 적은 곳에 사용함
- GPU 인스턴스는 정말로 비싸다.
 - 훈련 및 추론 최소 2개 필요
- Gen AI 의 임베딩은 CPU로도 괜찮다.

관리형 스팟 훈련

- 훈련용 EC2 스팟 인스턴스 사용함
- 온-디맨드 인스턴스 90% 이상 절약함
- 스팟 인스턴스는 언제나 가로챌 수 있음.
 - S3로 모델 체크포인트를 사용하라.
 - 학습 훈련(training jobs)을 재실행하라.

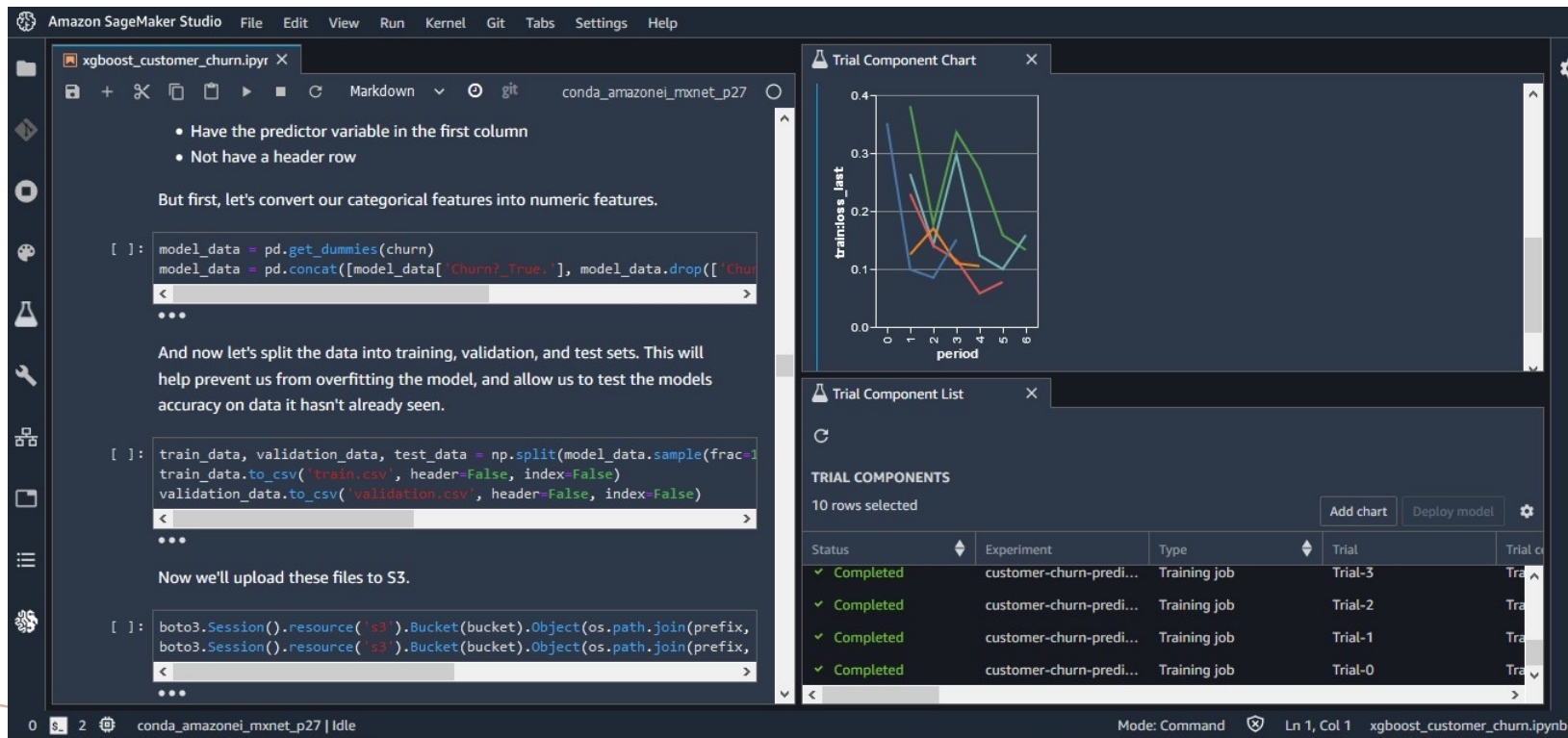




SAGEMAKER STUDIO 2022

SageMaker Studio

- 머신러닝용 비주얼 통합 개발 환경(IDE)
- 수많은 기능과 서비스 통합

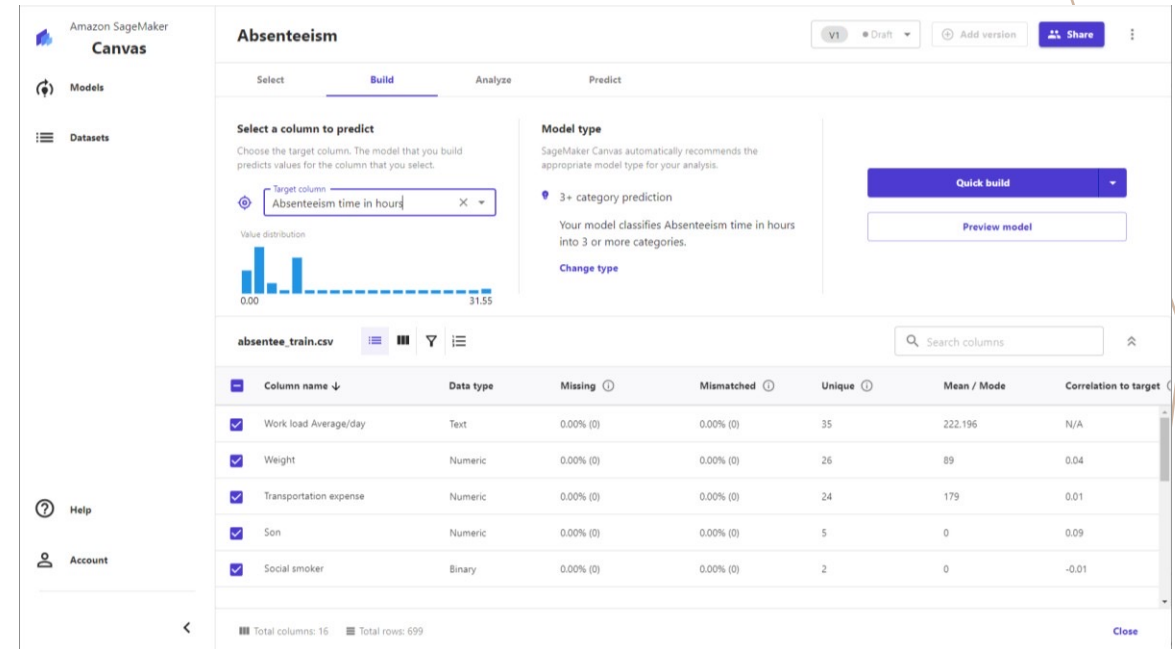


SageMaker 특징

- SageMaker JumpStart
 - model zoo 로 부터 모델과 알고리즘을 원-클릭으로 실행
 - NLP, 개체 감지, 이미지 분류 등 150개 이상 오픈-소스 모델
- SageMaker Data Wrangler
 - SageMaker Studio 에서 Import / transform / analyze / export 데이터
- SageMaker Feature Store
 - SageMaker Studio 에서 피처를 찾고, 발견 하고, 공유하기
 - 온라인 (낮은 대기 호출 시간) 또는 오프라인 (훈련 또는 배치 추론) 모드
 - 피처 그룹 안에 잘 정리된 피처
- SageMaker Edge Manager
 - 에지 장치용 소프트웨어 에이전트
 - 세이지메이커 니오로 최적화된 모델
 - 모니터링, 레이블링, 재훈련용 샘플 데이터 및 수집
- 비동기적인 추론 엔드포인트

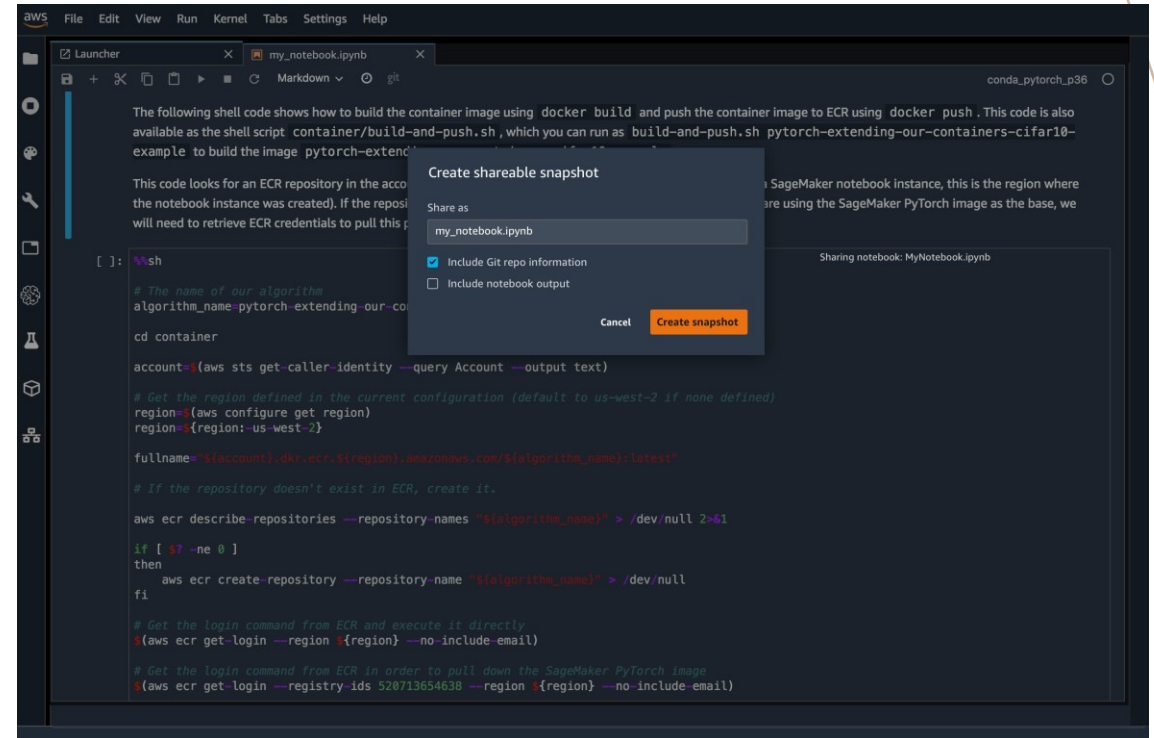
SageMaker Canvas

- 비즈니스 분석가용 노-코드 머신러닝
- csv 데이터 업로드
- 예측할 컬럼 선택하고, 빌드하고, 예측 생성.
- 데이터셋 결합(join)할 수 있음.
- 분류(Classification) 또는 회귀(regression)
- 자동 데이터 클리닝
 - 결측값(Missing values)
 - 아웃라이어
 - 중복 제거
- SageMaker Studio 와 함께 모델과 데이터셋 공유



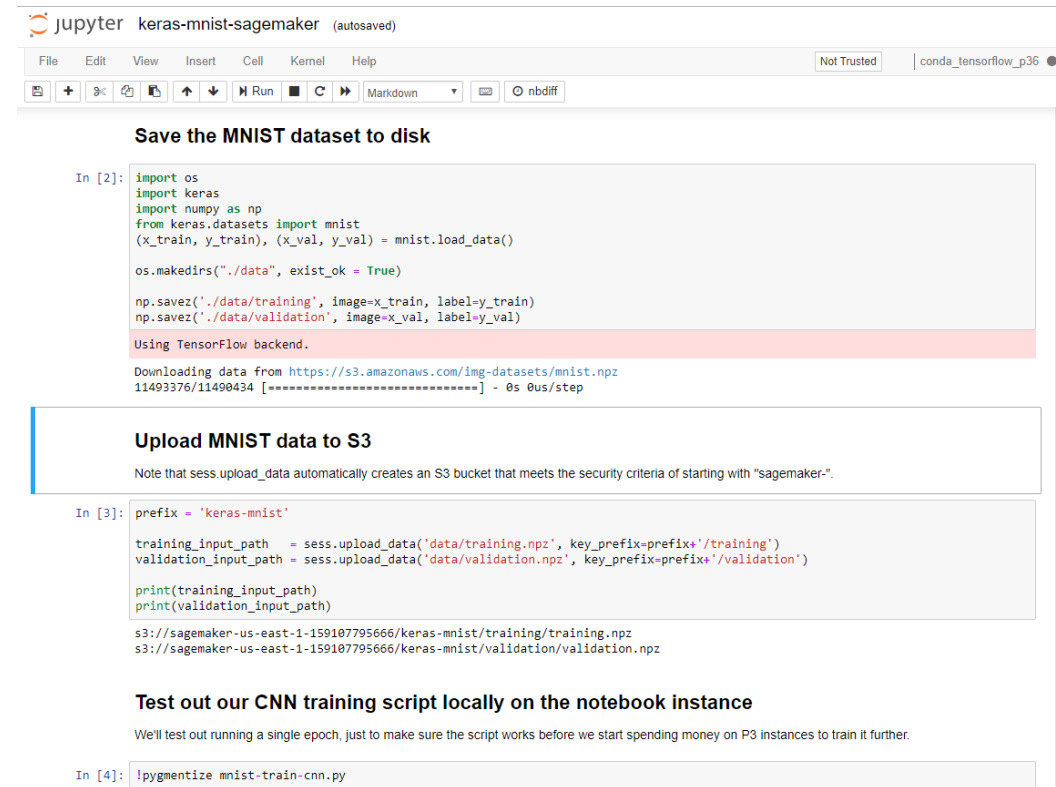
SageMaker Notebooks (1)

- SageMaker Studio 에서 Jupyter notebooks 생성 및 공유
- (인프라스트럭처 구성 없이 관리할 수 있는) 하드웨어 설정 스위치



SageMaker Notebooks (2)

- EC2 상에서 노트북 인스턴스들은 콘솔에서 실행
 - S3 데이터 접근
 - Scikit_learn, Spark, Tensorflow 지원
 - 다양한 빌트-인 모델 지원
 - 학습에 필요한 훈련 인스턴스 사용
 - 크기 조절을 할 수 있는 예측용 모델 배포



The screenshot shows a Jupyter Notebook interface with the title 'keras-mnist-sagemaker (autosaved)'. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Help) and a toolbar with icons for file operations and execution. The notebook content is divided into sections:

- Save the MNIST dataset to disk**: This section contains a code cell (In [2]) that imports necessary libraries and downloads the MNIST dataset. The code is as follows:

```
import os
import keras
import numpy as np
from keras.datasets import mnist
(x_train, y_train), (x_val, y_val) = mnist.load_data()

os.makedirs("./data", exist_ok = True)

np.savez('./data/training', image=x_train, label=y_train)
np.savez('./data/validation', image=x_val, label=y_val)

Using TensorFlow backend.

Downloading data from https://s3.amazonaws.com/img-datasets/mnist.npz
11493376/11490434 [=====] - 0s 0us/step
```
- Upload MNIST data to S3**: This section contains a code cell (In [3]) that uploads the saved data to an S3 bucket. The code is as follows:

```
prefix = 'keras-mnist'

training_input_path = sess.upload_data('data/training.npz', key_prefix=prefix+'/training')
validation_input_path = sess.upload_data('data/validation.npz', key_prefix=prefix+'/validation')

print(training_input_path)
print(validation_input_path)

s3://sagemaker-us-east-1-159107795666/keras-mnist/training/training.npz
s3://sagemaker-us-east-1-159107795666/keras-mnist/validation/validation.npz
```
- Test out our CNN training script locally on the notebook instance**: This section contains a code cell (In [4]) that runs a command to lint the training script.

```
!pygmentize mnist-train-cnn.py
```

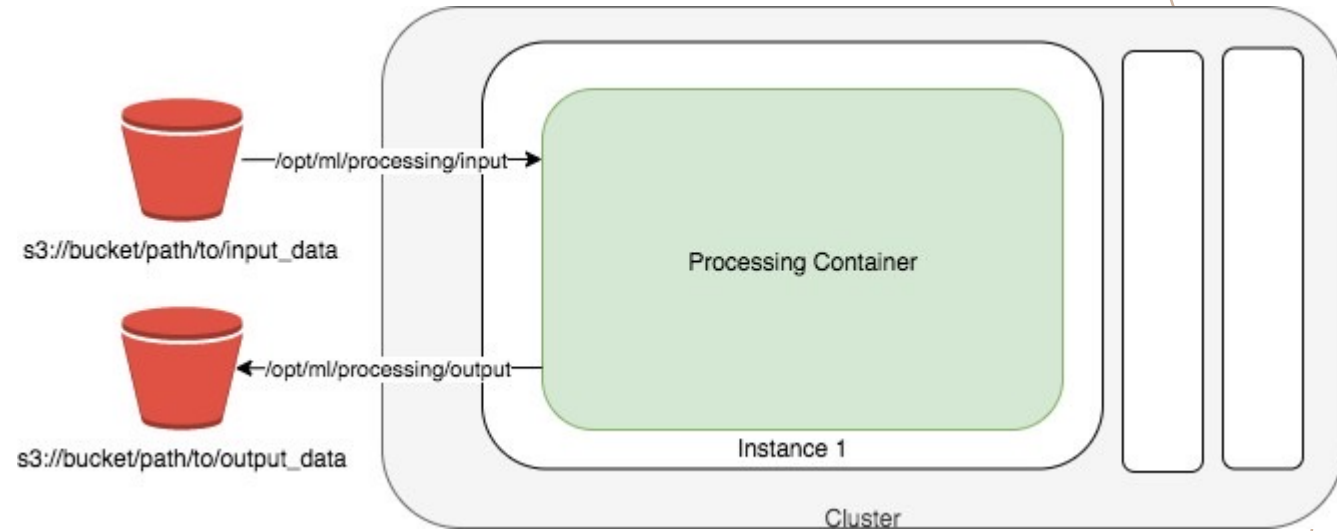
세이지메이커에서 데이터 사전 준비

- 데이터셋은 언제나 S3에서!
 - 모델 훈련은 Protobuf RecordIO 포맷을 주로 사용
 - 파이썬은 변수 그대로 사용하나 자바나 다른 언어는 스키마 제공해야 함.
 - RecordIO 포맷: 데이터셋의 각 관측치를 4바이트 부동 소수점 집합로 나타낸 이진 표현으로 변환한 다음 protobuf values 필드로 로드
- Athena, EMR, Redshift 및 Amazon Keyspaces D Postgre(pgvector) 에서 데이터 주입
- 세이지메이커와 아파치 스파크 통합
- 노트북에서 Scikit_learn, numpy, pandas 모두 설치 및 호출



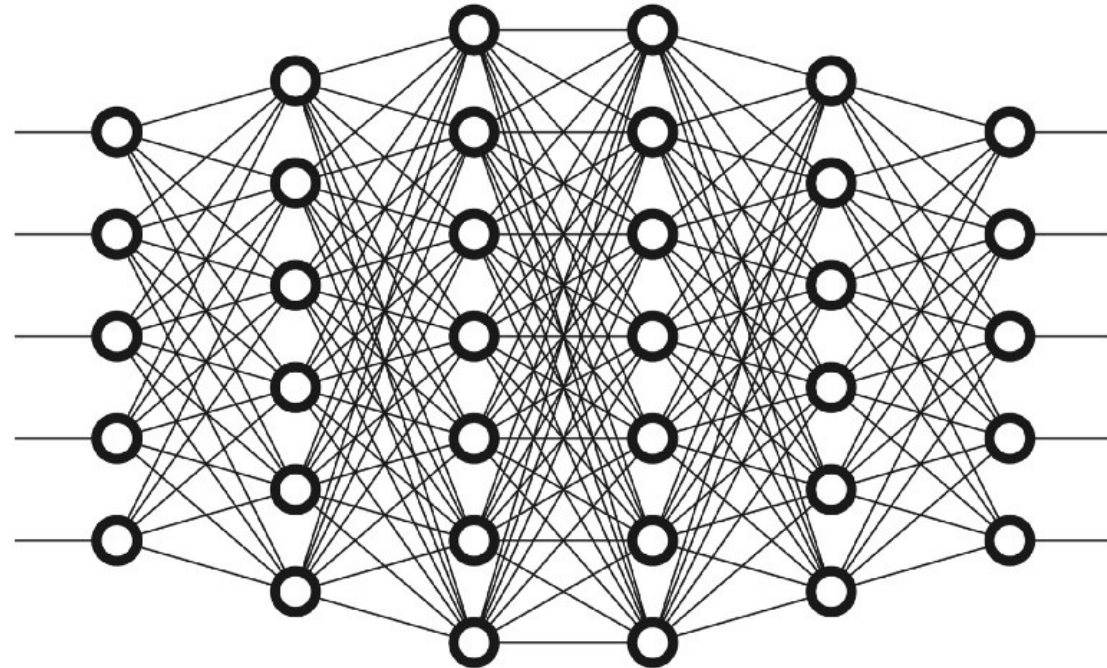
세이지메이커 프로세싱

- 프로세싱 잡스(Processing jobs) -> ECS와 거의 유사
 - S3로부터 데이터 복제
 - 프로세싱 컨테이너에서 동작함
 - 빌트-인 세이지메이커 또는 사용자 커스텀 지원
 - S3로 데이터 결과 처리



SageMaker Training 특징

- 하나의 Training Job 생성
 - 훈련 데이터를 위한 S3 버킷의 URL
 - ML 컴퓨트 리소스
 - 출력 결과용 S3 버킷 URL
 - ECR 패스를 훈련 코드로
- 훈련 옵션
 - 빌트-인 훈련 알고리즘
 - 스파크 MLlib
 - 커스텀 파이썬 텐서플로 / MXNet 코드
 - PyTorch, Scikit-Learn, RLEstimator
 - XGBoost, Hugging Face, Chainer
 - 여러분이 만든 도커 이미지
 - AWS 마켓플레이스에서 구매한 알고리즘



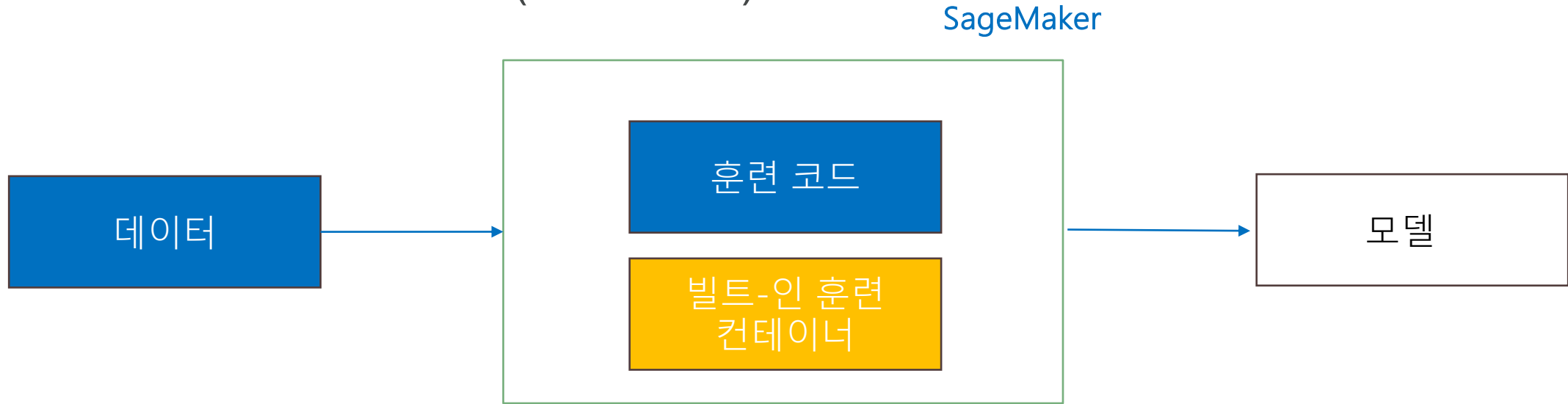
SageMaker Training (1)



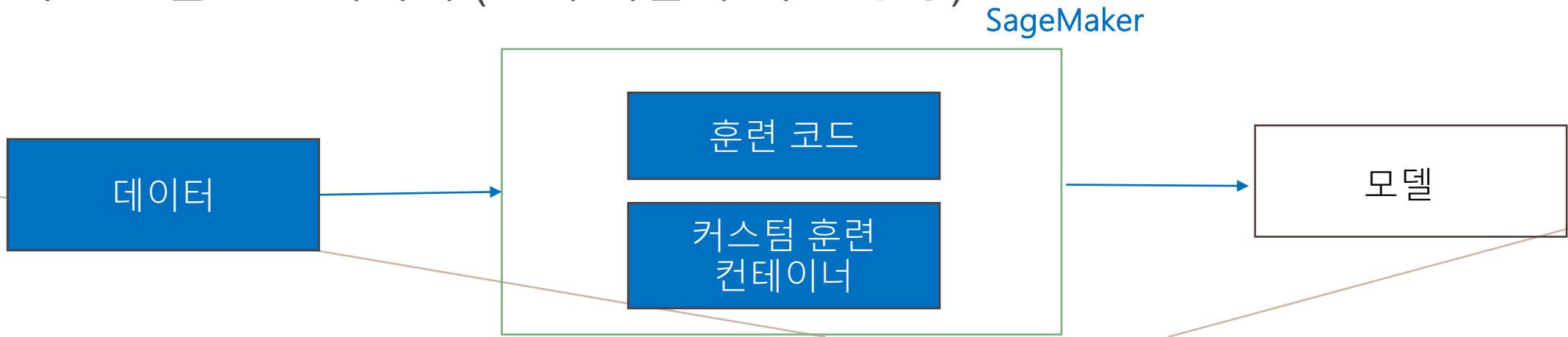
- 완전 관리형 머신러닝 훈련 프로세싱 잡스
- 데이터 과학자가 빠르고 쉽게 모델 개발, 빌드 및 훈련을 할 수 있도록 지원

SageMaker Training (2)

- 빌트-인 훈련 컨테이너 (AWS 제공)

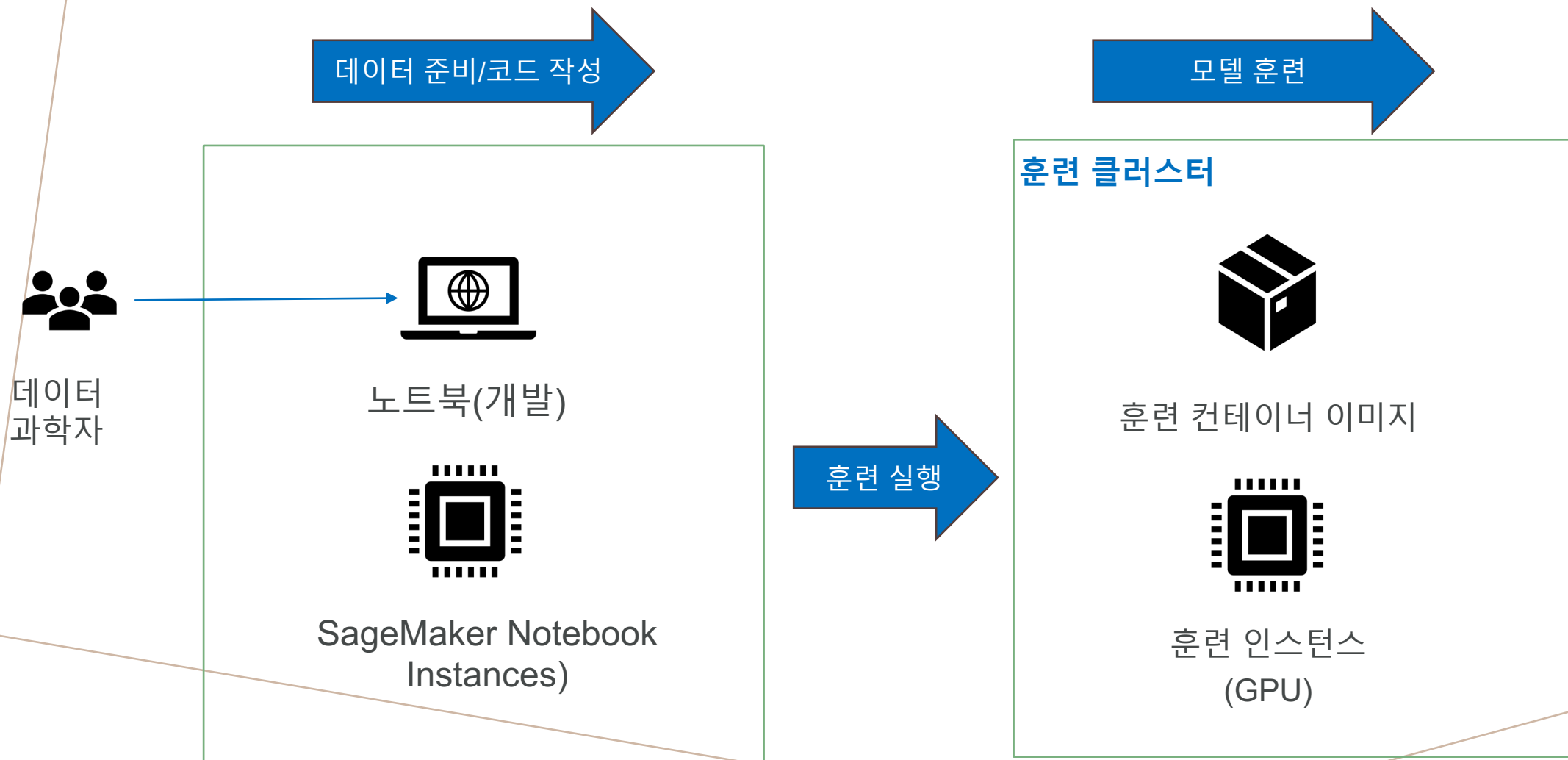


- 커스텀 훈련 컨테이너 (고객/개발자 직접 생성)



SageMaker Training (3)

- 훈련은 노트북 인스턴스 보다 GPU 인스턴스에서 수행하는 것을 권장



SageMaker Experiments (1)



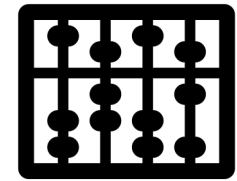
대규모 실험 추적
(Experiment Tracking)



실험 구조화
(Experiments 와 Trials)



실험 간 비교
(각 실험 간 시각화 및 비교)



평가 지수 기록
(SageMaker SDK 와 API를 사용해 맞춤 평가 지표를 기록하고 추적)

- 실험(Experiment): 시도 집합(Processing Jobs, Training Jobs, Batch Transform Jobs)
- 시도(Trial): 데이터셋, 알고리즘, 파라미터, 모델, 평가지수, 체크포인트 포함

SageMaker Experiments (2) – 코드 샘플

Experiment
생성

```
from smexperiments.experiment import Experiment
from smexperiments.trial import Trial

experiment = Experiment.create(
    experiment_name="experiment_name",
    description="Classification of mnist hand-written digits"
)
```

Trial 생성

```
trial = Trial.create(
    trial_name="trial_name",
    experiment_name=experiment.experiment_name,
    sagemaker_boto_client=sm,)
```

fit()

```
channel_name = "training"
estimator.fit(
    inputs={channel_name : data_path},
    job_name=job_name,
    experiment_config={
        "TrialName": trial.trial_name,
        "TrialComponentDisplayName": "Training"})
```



EXPERIMENTS	
0 row selected	
Create Experiment	
Name	Last Modified
Unassigned trial components	
8 minutes ago	
mnist-hand-written-digits-c...	32 minutes ago

TRIALS	
0 row selected	
Name	Last Modified
cnn-training-job-10-hidden-cha...	41 minutes ago
cnn-training-job-5-hidden-chan...	45 minutes ago
cnn-training-job-2-hidden-chan...	50 minutes ago

SageMaker Experiments(3) – 평가 지수 정의

Status	Experiment name	Trial name	mAP@.5	Precision	Recall	mAP@.5:.95
✓ Completed	yolov5-poc-exp1	yolov5-poc-exp1-p...	0.683	0.835	0.638	0.483
✓ Completed	yolov5-poc-exp1	yolov5-poc-exp1-p...	0.18	0.841	0.186	0.0703
✓ Completed	yolov5-poc-exp1	yolov5-poc-exp1-p...	0.18	0.841	0.186	0.0702
✓ Completed	yolov5-poc-exp1	yolov5-poc-exp1-p...	0.179	0.843	0.184	0.0701
✓ Completed	yolov5-poc-exp1	yolov5-poc-exp1-p...	0.179	0.84	0.186	0.0699

SageMaker Experiments 정보

Metric definitions에서 hooking된 정보

hyperparameters 정보

batch-size	cfg	data	epochs	freeze
128	"yolov5x.yaml"	"data/sm_dist_coco128..."	10	10
128	"yolov5s.yaml"	"data_sm.yaml"	10	10
128	"yolov5s.yaml"	"data_sm.yaml"	10	10
128	"yolov5s.yaml"	"data_sm.yaml"	10	10
128	"yolov5s.yaml"	"data_sm.yaml"	10	10

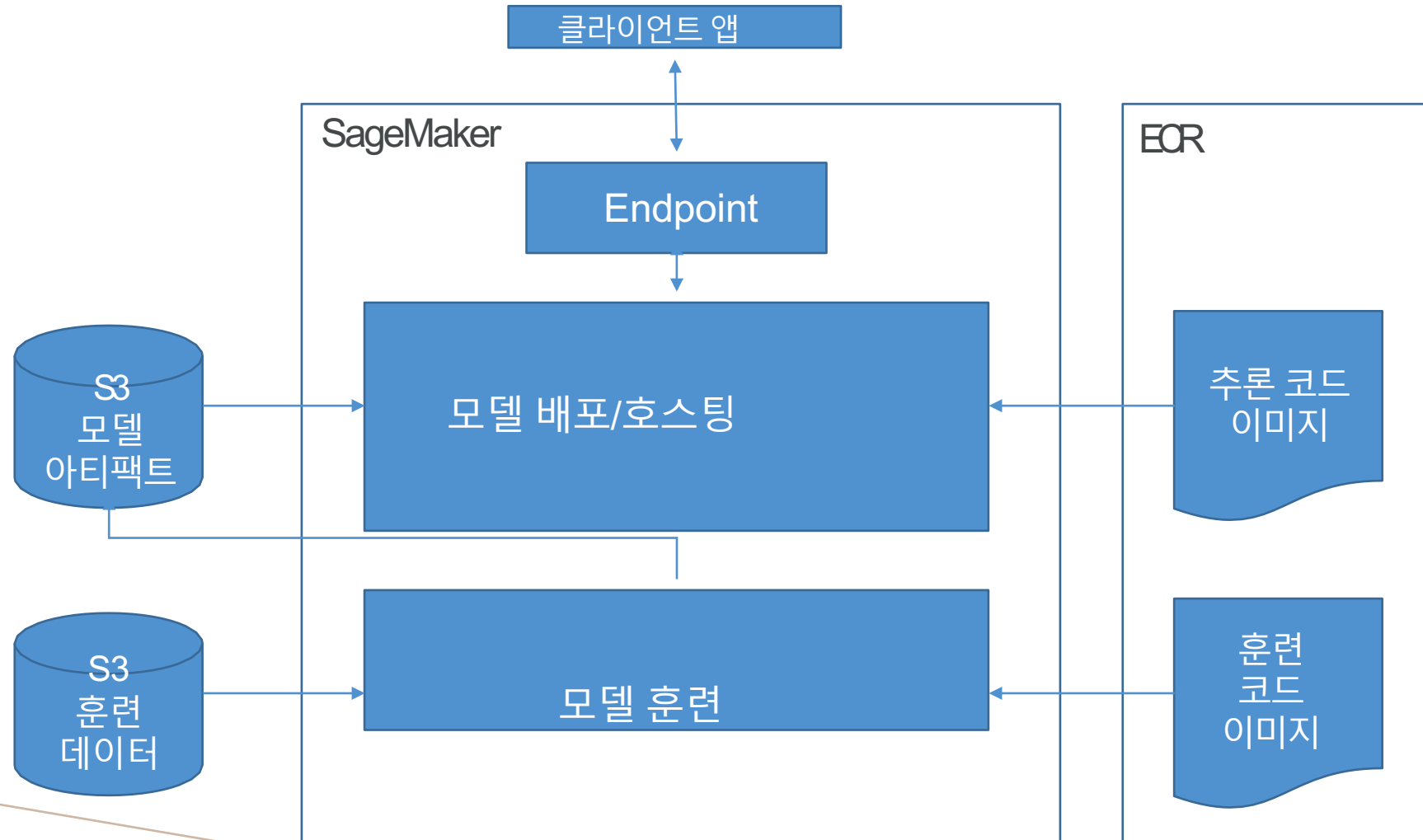


훈련한 모델 배포 (1)

- 훈련한 모델을 S3에 저장하기
- 두가지 형태로 배포할 수 있음:
 - 지속적인 엔드포인트로 개별적 예측 생성 (실시간 디맨드)
 - 전체 데이터셋용 예측할 수 있는 세이지메이커 배치 트랜스폼 (SageMaker Batch Transform)
- 모델 배포 옵션
 - 복잡한 프로세싱용 추론 파이프라인(Inference Pipelines) 지원
 - 에지(IoT) 장치를 이용한 세이지메이커 니오
 - 딥러닝 모델을 확장 가속화하는 탄력적 추론
 - 자동 크기 조정 (필요한 만큼 # 개 엔드포인트 증가/감소)
 - 현재 배포된 모델과 비교해 새 모델을 평가할 수 있는 쉐도우 테스트(Shadow Testing) 지원



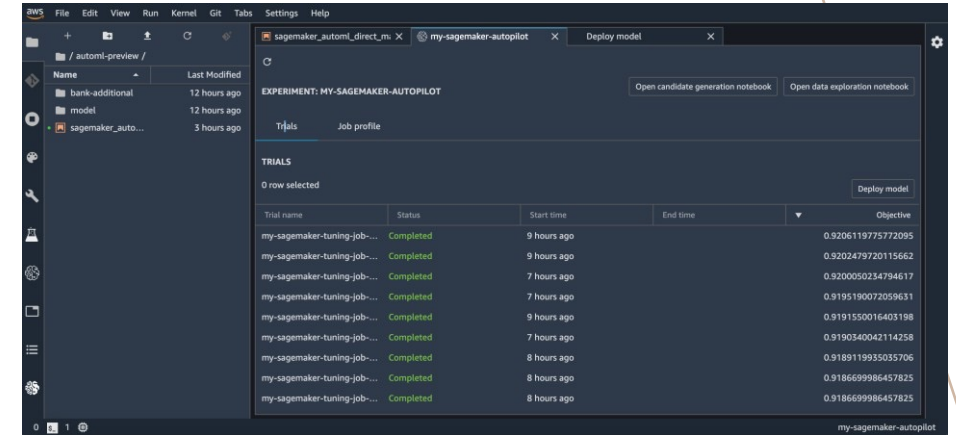
세이지메이커 훈련 및 배포 (2)



SageMaker Autopilot

SageMaker Autopilot 특징

- 자동화:
 - 알고리즘 선택
 - 데이터 전처리
 - 모델 튜닝
 - 모든 인프라스트럭처 자동 설정
- 시행착오를 줄여 줌
- AutoML



The screenshot shows the SageMaker Autopilot interface. On the left, a file explorer shows the project structure: / autopilot-preview /, with subfolders bank-additional, model, and sagemaker_auto... The main panel displays the 'EXPERIMENT: MY-SAGEMAKER-AUTOPILOT' page. It has tabs for 'Trials' and 'Job profile'. The 'Trials' tab is active, showing a table of completed tuning jobs. The table has columns for Trial name, Status, Start time, End time, and Objective. All jobs are marked as 'Completed' and occurred within the last 9 hours. A 'Deploy model' button is visible in the top right of the trials section.

Trial name	Status	Start time	End time	Objective
my-sagemaker-tuning-job-...	Completed	9 hours ago		0.9206119775772095
my-sagemaker-tuning-job-...	Completed	9 hours ago		0.9202479720115662
my-sagemaker-tuning-job-...	Completed	7 hours ago		0.9200050234794617
my-sagemaker-tuning-job-...	Completed	7 hours ago		0.9195190072059631
my-sagemaker-tuning-job-...	Completed	9 hours ago		0.9191550016403198
my-sagemaker-tuning-job-...	Completed	7 hours ago		0.9190340042114258
my-sagemaker-tuning-job-...	Completed	8 hours ago		0.9189119935035706
my-sagemaker-tuning-job-...	Completed	8 hours ago		0.9186699986457825
my-sagemaker-tuning-job-...	Completed	8 hours ago		0.9186699986457825

SageMaker Autopilot 워크플로

- S3 에서 훈련용 데이터 로드
- 예측을 위한 타겟 컬럼 선택
- 자동 모델 생성
- 모델 노트북은 눈에 보이고 조정 가능함.
- 모델 리더보드(Model leaderboard)
 - 모델 추천의 랭크 리스트
 - 그 중에 하나를 선택할 수 있음
- 필요하다면 노트북으로 모델을 배포 및 모니터링, 재정의 가능

Do you want to run a complete experiment?



Yes



No, run a pilot to create a notebook with candidate definitions

SageMaker Autopilot

- 인간 지도 추가 가능함.
- SageMaker Studio 또는 AWS SDK로 코드로 작성하거나 코드 없이 작성할 수 있음.
- 문제 형식 정의:
 - 이진 분류(Binary classification)
 - 다중 분류(Multiclass classification)
 - 회귀(Regression)
- 알고리즘 형식:
 - Linear Learner
 - XGBoost
 - Deep Learning (MLP – 멀리 레이어 퍼셉션)
 - 앙상블 모드
- 표형식의 CSV 또는 파케이(Parquet) 데이터

Autopilot Explainability

- SageMaker Clarify와 통합
- 어떻게 모델들이 예측값에 가까이 가는지 투명하게 보여줌
- 피쳐 속성
 - SHAP 베이스라인 / 균형 잡힌 값 사용
 - 협동 게임 이론으로부터 리서치
 - 지정한 예측용으로 각 피쳐를 할당

A Unified Approach to Interpreting Model Predictions

Scott M. Lundberg
Paul G. Allen School of Computer Science
University of Washington
Seattle, WA 98105
slund1@cs.washington.edu

Su-In Lee
Paul G. Allen School of Computer Science
Department of Genome Sciences
University of Washington
Seattle, WA 98105
suinlee@cs.washington.edu

Abstract

Understanding why a model makes a certain prediction can be as crucial as the prediction's accuracy in many applications. However, the highest accuracy for large modern datasets is often achieved by complex models that even experts struggle to interpret, such as ensemble or deep learning models, creating a tension between *accuracy* and *interpretability*. In response, various methods have recently been proposed to help users interpret the predictions of complex models, but it is often unclear how these methods are related and when one method is preferable over another. To address this problem, we present a unified framework for interpreting predictions, SHAP (SHapley Additive exPlanations). SHAP assigns each feature an importance value for a particular prediction. Its novel components include: (1) the identification of a new class of additive feature importance measures, and (2) theoretical results showing there is a unique solution in this class with a set of desirable properties. The new class unifies six existing methods, notable because several recent methods in the class lack the proposed desirable properties. Based on insights from this unification, we present new methods that show improved computational performance and/or better consistency with human intuition than previous approaches.

1 Introduction

The ability to correctly interpret a prediction model's output is extremely important. It engenders appropriate user trust, provides insight into how a model may be improved, and supports understanding of the process being modeled. In some applications, simple models (e.g., linear models) are often preferred for their ease of interpretation, even if they may be less accurate than complex ones. However, the growing availability of big data has increased the benefits of using complex models, so bringing to the forefront the trade-off between accuracy and interpretability of a model's output. A wide variety of different methods have been recently proposed to address this issue [5, 8, 9, 3, 4, 1]. But an understanding of how these methods relate and when one method is preferable to another is still lacking.

Here, we present a novel unified approach to interpreting model predictions.¹ Our approach leads to three potentially surprising results that bring clarity to the growing space of methods:

SageMaker Debugger and Monitor

SageMaker Debugger

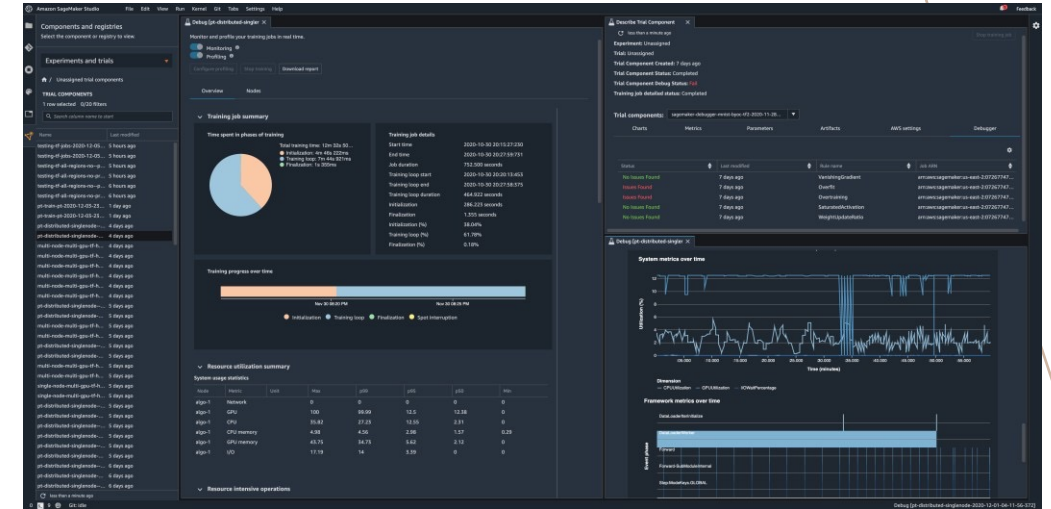
- 주기적인 간격에 내부 모델 상태 저장
 - 하나의 모델을 훈련하면, 기울기/텐서 간격 저장
 - 학습 훈련 동안 원치않는 조건을 감지해서 규칙 정의
 - 디버그 잡은 여러 분이 설정한 각 규칙을 실행
 - 규칙이 적용될 때 클라우드워치에서 로그 찍음
- SageMaker Studio 디버거 대시보드 지원
- 자동-생성된 훈련 리포트
- 빌트-인 규칙:
 - 시스템 병목현상 모니터링
 - 모델 프레임워크 오퍼레이션 프로파일
 - 모델 파라미터 디버깅

SageMaker Debugger

- 프레임워크 및 알고리즘 지원:
 - Tensorflow
 - PyTorch
 - MXNet
 - XGBoost
 - SageMaker generic estimator (커스텀 훈련 컨테이너 용도)
- GitHub 에서 디버거 API 사용가능
 - CreateTrainingJob과 DescribeTrainingJob API 용 후크 & 규칙 생성
 - SMDebug 클라이언트 라이브러리는 훈련 데이터를 접근하는 후크용으로 등록해서 사용 가능함.

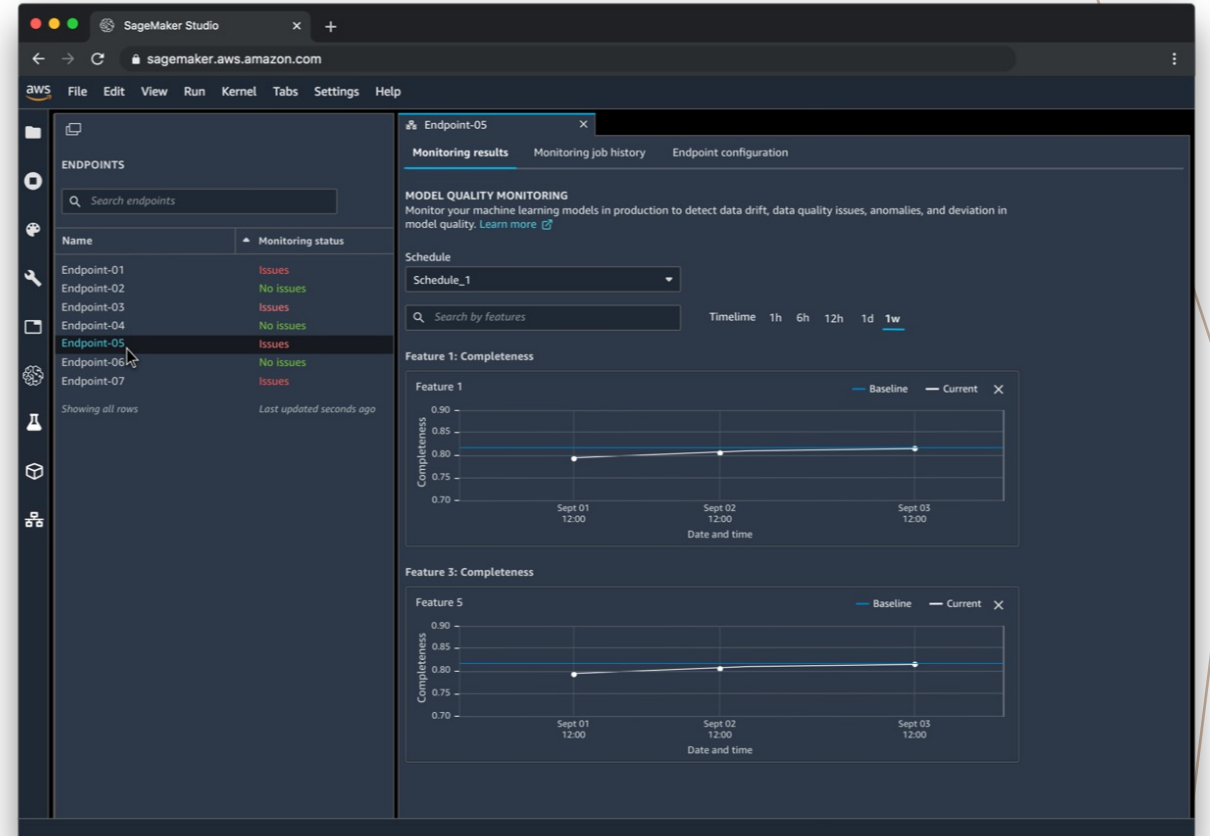
SageMaker Debugger

- SageMaker Debugger Insights Dashboard
- Debugger ProfilerRule
 - ProfilerReport
 - 하드웨어 시스템 평가지수 (CPUBottlenck, GPUMemoryIncrease 등)
 - 프레임워크 평가지수 (MaxInitializationTime, OverallFrameworkMetrics, StepOutlier)
- 알림을 받거나 훈련을 멈출 때 빌트-인 액션하기
 - StopTraining(), Email(), 또는 SMS()
 - Debugger 규칙에 대응
 - 알림 메시지 SNS로 보내기
- 시스템 리소스 사용과 훈련 프로파일링



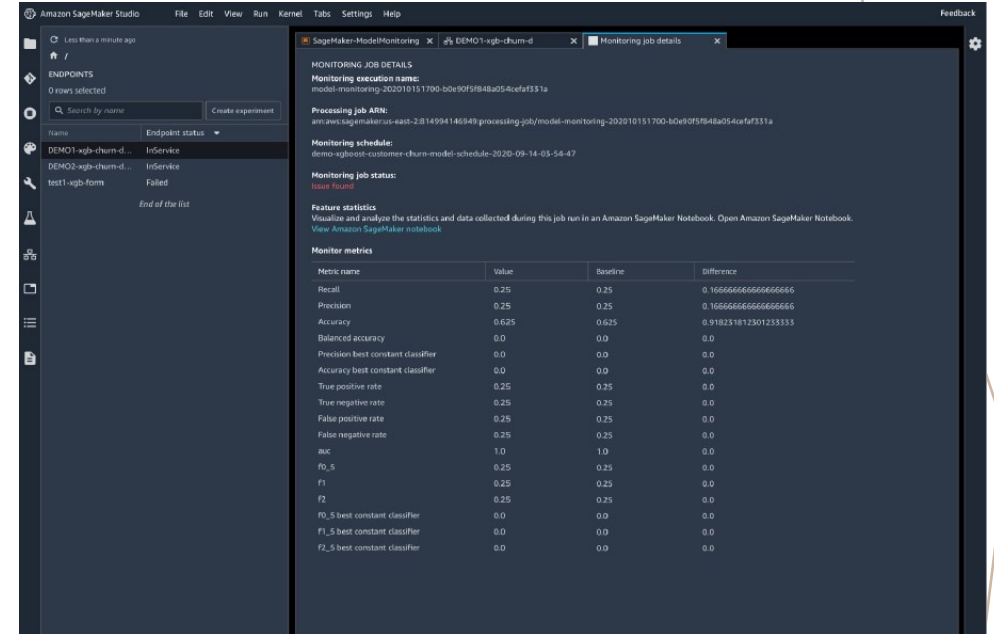
SageMaker Model Monitor

- 클라우드워치를 통해 여러분의 배포된 모델을 품질 탈선이 되면 경고 발생
- 데이터 드리프트 시각화
- 아웃라이어 및 이상 감지
- 새 피쳐 감지
- 코드가 필요 없음



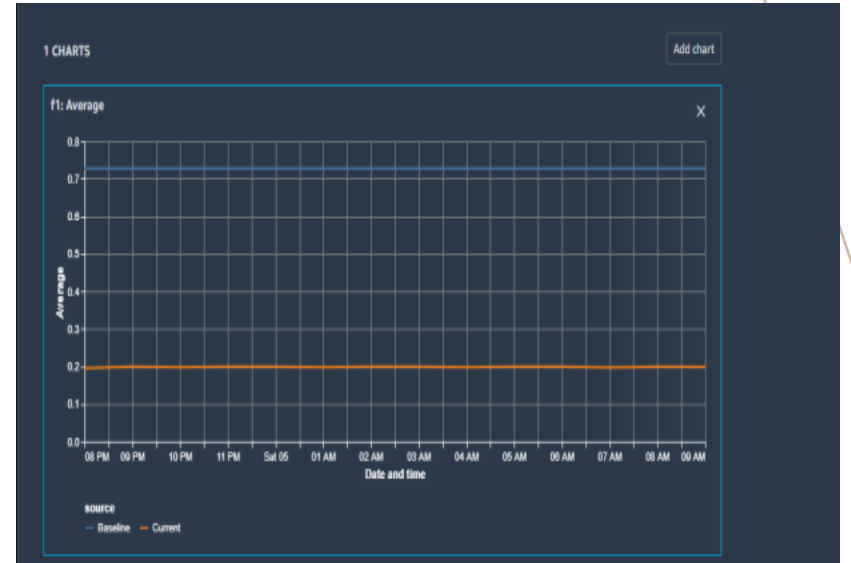
SageMaker Model Monitor + Clarify

- SageMaker Clarify 와 통합
 - SageMaker Clarify 는 잠재적인 편향 감지함
 - 다른 그룹/나이/불균형/소득 계층 간의 불균형
- ModelMonitor 과 함께, 편견을 모니터링하고 클라우드워치를 통해 새로운 잠재적인 편향을 경고함.
- SageMaker Clarify 는 모델의 여러가지 모습을 설명하는 데 도움을 줌.
 - 예측을 가장 잘 하는 피처를 살펴봄



SageMaker Model Monitor

- 데이터는 S3 에서 저장하고 안전하게 보호함
- Monitoring jobs 는 모니터링 스케줄을 통해 예약함
- 평가지수를 클라우드워치에 보냄.
 - CloudWatch 알림은 트리거 알람을 사용함.
 - 모델 교정 (모델 재훈련, 데이터 감사)
- Tensorboard, QuickSight, Tableau 와 통합
 - 또는 SageMaker Studio 에서 시각화



SageMaker Model Monitor

- 모니터링 타입:
 - 데이터 품질에서 드리프트
 - 여러분이 생성한 베이스라인과 연관
 - “품질”은 피처들의 통계 속성이다.
 - 모델 품질에서의 드리프트 (정확도)
 - 모델 품질 베이스라인과 똑같은 방식으로 동작함
 - Ground Truth 레이블과 통합
 - 편향 드리프트(Bias drift)
 - 피처 속성 드리프트
 - Normalized Discounted Cumulative Gain (NDCG) 스코어 기반
 - 훈련 데이터와 실제 데이터의 피처 랭킹 비교



Demo

```
$python train.py
```

Tensorflow

Keras, Scikit-learn, Pandas,
Numpy, Matplotlib, horovod

Python 3.10

GPU (CUDA 11.6 cuDNN 8.x)

컨테이너 런타임

NVIDIA / 호스트 OS

개발 서버

Push



ECR

Pull



```
$python train.py
```

Tensorflow

Keras, Scikit-learn, Pandas,
Numpy, Matplotlib, horovod

Python 3.10

GPU (CUDA 11.6 cuDNN 8.x)

컨테이너 런타임

NVIDIA / 호스트 OS

훈련, 추론, 운영 서버