

9. 세이지메이커 스튜디오 I

2강. 세이지메이커 스튜디오 모델 훈련

학습목표

- 세이지메이커 스튜디오에서 데이터 준비 및 구축 단계를 설명할 수 있다.
- 세이지메이커 스튜디오에서 모델 훈련 과정을 설명할 수 있다.

학습내용

- 데이터 준비 및 구축
- 모델 훈련

1. 데이터 준비 및 구축

- 데이터 준비 및 구축 단계
 - ① Studio에 로그인
 - ② 예제 소스 다운로드
 - ③ 소스파일 열기
 - ④ 커널 및 인스턴스 선택

✓ SSO를 사용하여 할당 된 링크로 Studio 액세스




✓ 예제 소스 다운로드

- 시스템 터미널 열기
- 예제 다운로드 명령어 입력
- `git clone https://github.com/aws-labs/amazon-sagemaker-examples.git`

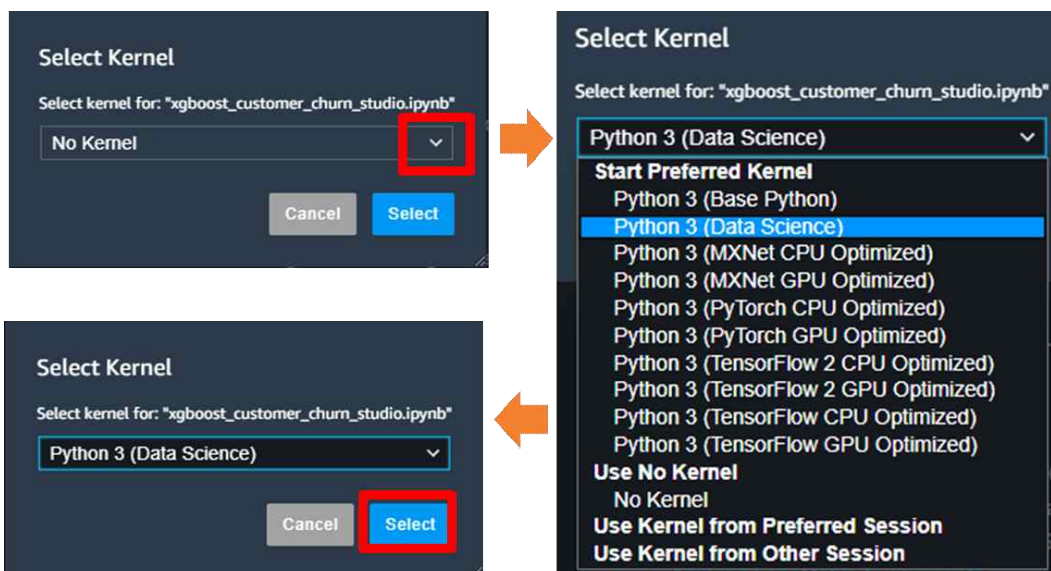


✓ 소스파일 열기

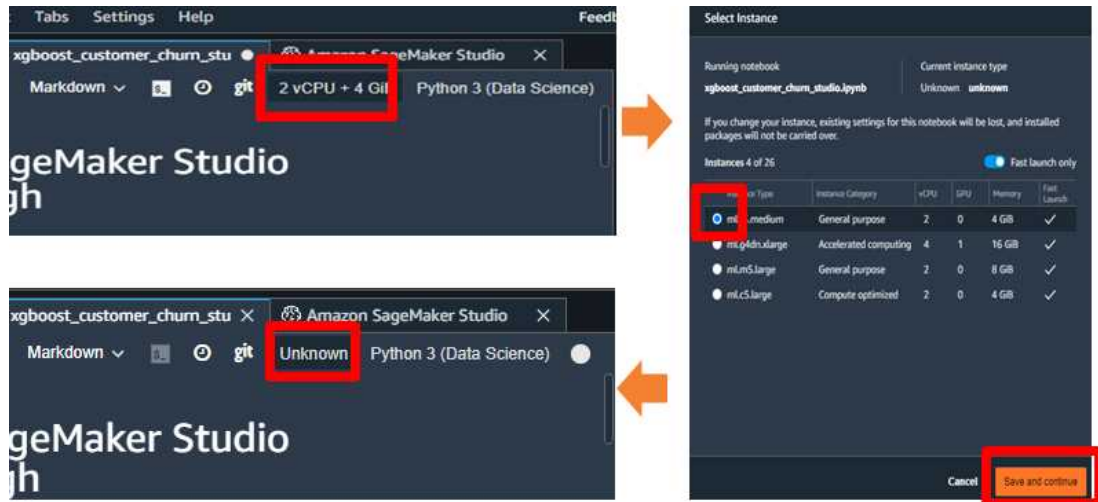
- 파일 브라우저 아이콘 () 선택
- 사전 준비된 파일을 통해 예제 열기
- 파일 경로 : `amazon-sagemaker-examples/aws_sagemaker_studio_getting_started`
- `xgboost_customer_churn_studio.ipynb` 두 번 클릭

✓ 커널 및 인스턴스 선택

- 커널 선택 Python3 (Data Science)



- 인스턴스 선택 (ml.t3.medium)



2. 모델 훈련

- 모델 훈련 단계

- ① 모델 훈련 준비
- ② 실험 생성
- ③ 모델 훈련
- ④ 훈련 결과 확인
- ⑤ 여러 시도(Trial) 생성 및 비교

- ✓ 모델 훈련 준비

- 필요한 라이브러리를 추가(import)

```
[ ]: import sys
      [sys.executable] # pip install sagemaker U
      [sys.executable] # pip install sagemaker experiments

[ ]: import pandas as pd
      import numpy as np
      import matplotlib.pyplot as plt
      import io
      import os
      import sys
      import time
      import json
      from IPython.display import display
      from time import strftime, gettime
      import boto3
      import re

      import sagemaker
      from sagemaker import get_execution_role
      from sagemaker.predictor import csv_serializer
      from sagemaker.debugger import rule_configs, Rule, DebuggerHookConfig
      from sagemaker.model_monitor import DataCaptureConfig, DatasetFormat, DefaultModel
      from sagemaker.i3 import S3Uploader, S3Downloader

      from smexperiments.experiment import Experiment
      from smexperiments.trial import Trial
      from smexperiments.trial_component import TrialComponent
      from smexperiments.tracker import Tracker

[ ]: sess = boto3.Session()
      sm = sess.client('sagemaker')
      role = sagemaker.get_execution_role()
```

- pandas를 사용하여 헤더가 있는 훈련 데이터의 일부를 로드
- 데이터를 탐색하여 데이터 집합의 기능과 모델 학습에 사용될 데이터를 확인

```
# Set the path we can find the data files that go with this notebook
%cd /root/amazon-sagemaker-examples/aws_sagemaker_studio/getting_started
local_data_path = './data/training-dataset-with-header.csv'
data = pd.read_csv(local_data_path)
pd.set_option('display.max_columns', 500)      # Make sure we can see all of
pd.set_option('display.max_rows', 10)         # Keep the output on one page
data
```

- 훈련을 위하여 s3에 데이터 업로드
- 데이터에 대한 s3 버킷이 없는 경우 s3 버킷 생성

```
account_id = sess.client('sts', region_name=sess.region_name).get_caller_identity()
bucket = 'sagemaker-studio-{}'.format(sess.region_name, account_id)
prefix = 'xgboost-churn'

try:
    if sess.region_name == 'us-east-1':
        sess.client('s3').create_bucket(Bucket=bucket)
    else:
        sess.client('s3').create_bucket(Bucket=bucket,
                                         CreateBucketConfiguration={'LocationConstraint': sess.region_name})
except Exception as e:
    print("Looks like you already have a bucket of this name. That's good. Uploading")

# Return the URLs of the uploaded file, so they can be reviewed or used elsewhere
s3url = S3Uploader.upload('data/train.csv', 's3://{}{}'.format(bucket, prefix),
                           print(s3url))
s3url = S3Uploader.upload('data/validation.csv', 's3://{}{}'.format(bucket, prefix),
                           print(s3url))
```

- 명령어 입력 후 s3에 버킷이 자동으로 생성됨

새로운 S3 콘솔 환경을 계속 개선하는 동안 이 버전의 S3 콘솔을 일시적으로 복원했습니다.

S3 버킷

버킷 검색

모든 액세스 유형

+ 버킷 만들기 퍼블릭 액세스 설정 편집 비우기 삭제 1 리전

1 버킷

<input type="checkbox"/>	버킷 이름	액세스	리전	생성 날짜
<input type="checkbox"/>	sagemaker-studio-us-east-1-40234501...	객체를 퍼블릭으로 설정할 수 있음	미국 동부(버 지니아 북부)	8월 1, 2020 4:51:24 오후 GMT+0900

- 생성된 버킷 안에는 xgboost-churn 폴더가 생성
- 폴더 안에는 train 데이터와 validation 데이터 폴더가 각각 생성



- 현재 실습 코드에서는 XGBoost 라이브러리를 사용하여 업로드 한 데이터의 모델 훈련을 진행하고자 함
- XGBoost 알고리즘 컨테이너의 위치를 지정

```
from sagemaker.amazon.amazon_estimator import get_image_uri
docker_image_name = get_image_uri(boto3.Session().region_name, 'xgboost')
```

- s3_inputs을 생성
- s3에 있는 데이터에 대한 포인터로서 훈련에 사용할 경로 설정



```
s3_input_train = sagemaker.s3_input(s3_data='s3://{}/{}/train'.format(bucket, prefix))
s3_input_validation = sagemaker.s3_input(s3_data='s3://{}/{}/validation/'.format(bucket, prefix))
```

✓ 모델 훈련 단계

- 모델 훈련 추적, 이전 모델과의 비교, 모델 구성, 매개변수(하이퍼파라미터), 매트릭스를 로그로 기록하고 비교하기 위하여 실험을 생성

```
sess = sagemaker.session.Session()

create_date = strftime("%Y-%m-%d-%H-%M-%S", gmtime())
customer_churn_experiment = Experiment.create(experiment_name="customer-churn-pr",
                                              description="Using xgboost to predict customer churn",
                                              sagemaker_boto_client=boto3.client('sagemaker'))
```

- 왼쪽 사이드바에서 실험 목록 아이콘 () 클릭
- 실험 목록 새로고침() 클릭
- 실험 목록에서 실험 확인
- 실험명 : customer-churn-prediction-xgboost...

Name	Last modified
<i>Unassigned trial components</i>	
customer-churn-prediction-xgboost-2020-07-30-10-56...	16 minutes ago
bdutest-2020-07-14-06-29-48	16 days ago

✓ 모델 훈련

- XGBoost 하이퍼파라미터를 지정

```
hyperparams = {"max_depth":5,
               "subsample":0.8,
               "num_round":600,
               "eta":0.2,
               "gamma":4,
               "min_child_weight":6,
               "silent":0,
               "objective":"binary:logistic"}
```

- XGBoost 알고리즘을 사용하여 이전 단계에서 생성된 실험에 할당된 시도 (Trial) 를 생성
- 훈련 작업을 시도(Trial) 구성 요소로 생성한 후 fit 메소드를 호출하여 시도(Trial) 를 실행
- 훈련 작업을 완료하는데 몇 분 정도 소요

```
trial = Trial.create(trial_name="algorithm-mode-trial-{}".format(strftime("%Y-%m-%d-%H-%M-%S", gmtime()))),
                  experiment_name=customer_churn_experiment.experiment_name,
                  sagemaker_boto_client=boto3.client('sagemaker'))

xgb = sagemaker.estimator.Estimator(image_name=docker_image_name,
                                     role=role,
                                     hyperparameters=hyperparams,
                                     train_instance_count=1,
                                     train_instance_type='ml.m4.xlarge',
                                     output_path='s3:///{}(/)/output'.format(bucket, prefix),
                                     base_job_name="demo-xgboost-customer-churn",
                                     sagemaker_session=session)

xgb.fit({'train': s3_input_train,
        'validation': s3_input_validation},
        experiment_config={
            "ExperimentName": customer_churn_experiment.experiment_name,
            "TrialName": trial.trial_name,
            "TrialComponentDisplayName": "Training",
        })
```


✓ 훈련 종료

```
[592]#011train-error:0.020574#011validation-error:0.061562
[593]#011train-error:0.020574#011validation-error:0.061562
[594]#011train-error:0.020574#011validation-error:0.061562
[595]#011train-error:0.020574#011validation-error:0.061562
[596]#011train-error:0.020574#011validation-error:0.061562
[597]#011train-error:0.020574#011validation-error:0.061562
[598]#011train-error:0.020574#011validation-error:0.061562
[599]#011train-error:0.021003#011validation-error:0.061562

2020-07-30 12:23:28 Uploading - Uploading generated training model
2020-07-30 12:23:28 Completed - Training job completed
Training seconds: 73
Billable seconds: 73
```

✓ 훈련 결과 확인

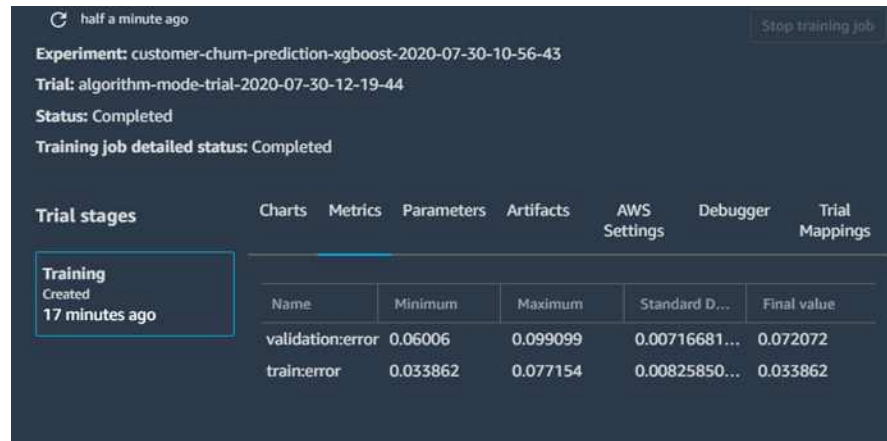
- 실험 목록에서 실험이름을 두 번 클릭하여시도(Trial) 확인
- 실험이름 →시도(Trial) 확인(algorithm-mode-trial...)

The first screenshot shows a table with columns 'Name' and 'Last modified'. A red box highlights the name 'customer-churn-predicti...'. A red callout box with the text '실험이름 더블 클릭' (Double-click experiment name) points to this entry.

The second screenshot shows a table with columns 'Name' and 'Last modified'. A red box highlights the name 'algorithm-mode-trial-20...'. A red callout box with the text '시도 더블 클릭' (Double-click trial) points to this entry.

The third screenshot shows a table with columns 'Name' and 'Last modified'. A red box highlights the name 'Training'. A red callout box with the text '더블 클릭' (Double-click) points to this entry.

- 훈련 시도 구성 요소(Train)를 두 번 클릭하여 구성 요소 설명 탭을 오픈



- 시도의 세부 정보 확인
- Describe Trial Component 메뉴에서 훈련 작업 시도 구성 요소에 대한 정보를 확인

Chart	고유한 차트 작성
Metrics	시도 실행 중 Tracker에서 기록되는 지표
Parameters	하이퍼파라미터 값 및 인스턴스 정보
Artifacts	입력 데이터 세트 및 출력 모델용 s3 스토리지
Aws setting	작업 이름, ARN, 상태, 생성 시간, 훈련 시간 등
Debugger	디버거 규칙 및 발견된 문제 목록
Trial Mappings	시도 매핑

✓ 여러 시도(Trial) 생성 및 비교

- 모델을 개선하려면 일반적으로 다른 하이퍼파라미터 값을 사용하여 최종 유효성 검사 오류에 영향을 미치는지 확인
- min_child_weight 파라미터를 변경하고 다른 값으로 다양한 훈련 작업을 시작하여 유효성 검사 오류에 미치는 영향을 살펴 봄
- 각각 서로 다른 min_child_weight 하이퍼파라미터 값을 사용하여 5가지 시도(Trial)를 생성하고 실행하는 셀을 실행
- 이전 시도에서 min_child_weight = 6
- 현재 시도에서는 min_child_weight = 1, 2, 4, 8, 10


```

min_child_weights = [1, 2, 4, 8, 10]

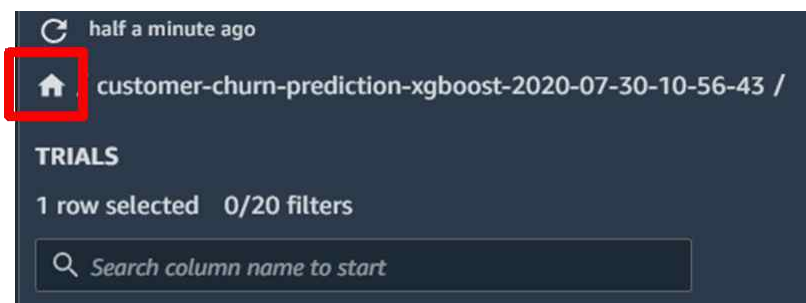
for weight in min_child_weights:
    hyperparams["min_child_weight"] = weight
    trial = Trial.create(trial_name="algorithm-xgboost-trial-{}".format(strftime("%Y-%m-%d-%H-%M-%S", gmtime()), weight),
                       experiment_name=customer_churn_experiment.experiment_name,
                       sagemaker_boto_client=boto3.client("sagemaker"))

    t_xgb = sagemaker.estimator.Estimator(image_name=docker_image_name,
                                          role=role,
                                          hyperparameters=hyperparams,
                                          train_instance_count=1,
                                          train_instance_type="ml.m4.xlarge",
                                          output_path="s3://{}{}".format(bucket, prefix),
                                          base_job_name="demo-xgboost-customer-churn",
                                          sagemaker_session=session)

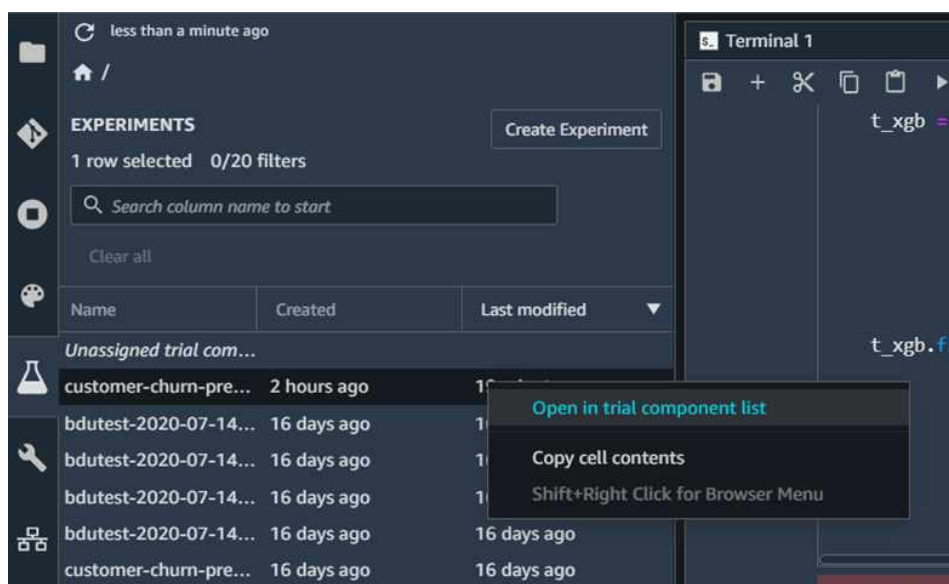
    t_xgb.fit({"train": s3_input_train,
              "validation": s3_input_validation},
             wait=False,
             experiment_config={
                 "ExperimentName": customer_churn_experiment.experiment_name,
                 "TrialName": trial.trial_name,
                 "TrialComponentDisplayName": "Training",
             })

```

- Studio에서 진행 상황을 추적하고 결과를 보려면 시도 구성 요소 위에 있는 홈 아이콘 선택



- 실험 이름을 마우스 오른쪽 버튼으로 클릭하고 시도 구성 요소 목록에서 열기 선택



- 시도 구성 요소 목록에서는 아래와 같은 기능을 수행할 수 있음
 - 시도에 대한 세부 정보를 확인
 - 시도를 비교하여 최고의 성과를 보이는 모델을 찾기
 - 훈련 결과를 시각화하기 위한 차트를 생성
- 시도 구성 요소 목록

less than 5 seconds ago

TRIAL COMPONENTS


1 row selected 0/20 filters

Search column name to start

Clear all

Status ▲	Experiment name	Trial name	Trial component name ▼
✓ Completed	customer-churn-predic...	algorithm-mode-trial-2...	Training
✓ Completed	customer-churn-predic...	algorithm-mode-trial-2...	Training
✓ Completed	customer-churn-predic...	algorithm-mode-trial-2...	Training
✓ Completed	customer-churn-predic...	algorithm-mode-trial-2...	Training
✓ Completed	customer-churn-predic...	algorithm-mode-trial-2...	Training
✓ Completed	customer-churn-predic...	algorithm-mode-trial-2...	Training
✓ Completed	customer-churn-predic...	algorithm-mode-trial-2...	Training

End of the list

- 오른쪽 상단 모서리에 설정 아이콘 ()클릭
- Summary에서 Trial name 클릭
- Metrics 전체 클릭



- min_child_weight 하이퍼파라미터 값에 대한 validation:error, train:error 비교 가능

Trial name	validation:error	train:error
algorithm-mode-trial-2020-07-30-12-49-46-weight-10	0.075075	0.042006
algorithm-mode-trial-2020-07-30-12-52-56	0.072072	0.036434
algorithm-mode-trial-2020-07-30-12-19-44	0.072072	0.033862
algorithm-mode-trial-2020-07-30-12-49-40-weight-4	0.06006	0.032576
algorithm-mode-trial-2020-07-30-12-49-38-weight-1	0.06006	0.030862
algorithm-mode-trial-2020-07-30-12-49-38-weight-2	0.064565	0.024432
algorithm-mode-trial-2020-07-30-12-49-44-weight-8	0.06606599688529968	0.02100300043821335

평가하기

1. 세이지메이커 스튜디오에서 구축 단계로 볼 수 있는 것은?

- ① Studio에 로그인
- ② 예제 소스 다운로드
- ③ 소스파일 열기
- ④ 커널 및 인스턴스 선택

- 정답 : ④번

해설 : 세이지메이커 스튜디오에서는 커널 및 인스턴스 선택을 통해 개발 환경을 빠르고 쉽게 구축할 수 있습니다.

2. 세이지메이커 스튜디오의 모델 훈련 과정에서 필요한 라이브러리를 추가하고 S3에 데이터를 업로드 하는 단계로 볼 수 있는 것은?

- ① 모델 훈련 준비
- ② 실험 생성
- ③ 모델 훈련
- ④ 훈련 결과 확인

- 정답 : ①번

해설 : 모델 훈련 준비 단계에서는 필요한 라이브러리 추가, 데이터 확인, S3에 데이터 업로드, XGBoost 알고리즘 컨테이너의 위치를 지정, S3_inputs 생성 등을 수행합니다.

학습정리

1. 데이터 준비 및 구축

- Studio에 로그인 : SSO를 사용하여 액세스
- 예제 소스 다운로드 : 시스템 터미널에서 예제 다운로드 명령어 입력
- 소스파일 열기 : 사전 준비된 파일을 통해 예제 열기
- 커널 및 인스턴스 선택 : Python3, ml.t3.medium

2. 모델 훈련

- 모델 훈련 준비 : 필요한 라이브러리 추가, 데이터 확인, S3에 데이터 업로드, XGBoost알고리즘 컨테이너의 위치를 지정, S3_inputs 생성
- 실험 생성 : 실험에 대한 로그를 기록하고 비교하기 위하여 실험을 생성
- 모델 훈련 : 하이퍼파라미터 지정, fit 메소드 호출
- 훈련 결과 확인 : 실험 목록에서 실험이름을두 번 클릭하여 시도(trial) 확인
- 여러 시도(Trial) 생성 및 비교 : 모델 학습을여러 번 시도하여 결과를 비교 분석