

10. 세이지메이커 스튜디오 표

3강. 세이지메이커 스튜디오 모델 실습

학습목표

- 세이지메이커 스튜디오에서 훈련이 종료된 모델의 결과를 시각화, 디버깅, 배포, 리소스 정리를 할 수 있다.

학습내용

- 훈련 결과 시각화
- 훈련 작업 디버깅
- 모델 배포
- 최상의 모델 배포
- 리소스 정리

1. 훈련 결과 시각화

- 차트화할 시도 선택
- ✓ [add chart] 선택

TRIAL COMPONENTS
5 rows selected 0/20 filters

Search column name to start

Clear all

Add chart Deploy model

Trial name	validation:error	train:error
algorithm-mode-trial-2020-08-01-09-00-00-weight-8	0.06606599688529968	0.02100300043821335
algorithm-mode-trial-2020-08-01-09-00-00-weight-10	0.066066	0.029147
algorithm-mode-trial-2020-08-01-08-59-59-weight-4	0.070571	0.046721
algorithm-mode-trial-2020-08-01-08-59-54-weight-1	0.069069	0.04072
algorithm-mode-trial-2020-08-01-08-59-54-weight-2	0.064565	0.025289
algorithm-mode-trial-2020-08-01-08-26-45	0.067568	0.033005

End of the list

- 하단의 new 차트 선택

TRIAL COMPONENTS 5 rows selected. Select rows to toggle chart visibility.

	Experiment	Trial	Trial Component	Type	Training time	Actions
Ⓢ	customer-churn-predic...	algorithm-mode-trial-2...	demo-xgboost-customer-churn-2020-08-01-09-00-00-644-aws-trai...	Training job	~1 minute	Remove
Ⓢ	customer-churn-predic...	algorithm-mode-trial-2...	demo-xgboost-customer-churn-2020-08-01-09-00-01-062-aws-trai...	Training job	~1 minute	Remove
Ⓢ	customer-churn-predic...	algorithm-mode-trial-2...	demo-xgboost-customer-churn-2020-08-01-08-59-59-215-aws-trai...	Training job	~1 minute	Remove
Ⓢ	customer-churn-predic...	algorithm-mode-trial-2...	demo-xgboost-customer-churn-2020-08-01-08-59-54-773-aws-trai...	Training job	~1 minute	Remove
Ⓢ	customer-churn-predic...	algorithm-mode-trial-2...	demo-xgboost-customer-churn-2020-08-01-08-59-55-065-aws-trai...	Training job	~1 minute	Remove

1 CHART

New Chart

Configure the new chart using the chart preferences panel.
Can't see the preferences panel? To open it click here.

- 차트 속성 구성

TRIAL COMPONENTS 5 rows selected. Select rows to toggle chart visibility.

	Experiment	Trial	Trial Component	Type	Training time
Ⓢ	customer-churn-predic...	algorithm-mode-trial-2...	demo-xgboost-customer-c...	Training job	~1 minute
Ⓢ	customer-churn-predic...	algorithm-mode-trial-2...	demo-xgboost-customer-c...	Training job	~1 minute
Ⓢ	customer-churn-predic...	algorithm-mode-trial-2...	demo-xgboost-customer-c...	Training job	~1 minute
Ⓢ	customer-churn-predic...	algorithm-mode-trial-2...	demo-xgboost-customer-c...	Training job	~1 minute
Ⓢ	customer-churn-predic...	algorithm-mode-trial-2...	demo-xgboost-customer-c...	Training job	~1 minute

1 CHART

New Chart

Configure the new chart using the chart preferences panel.
Can't see the preferences panel? To open it click here.

CHART PROPERTIES

Data type

- ☐ Time series
- ☒ Summary statistics

Chart type

- ☒ Histogram
- ☐ Line

X-axis dimension

- ☐ Epoch
- ☐ Time
- ☐ Periods from start

X-axis aggregation

- ☐ 1-minute
- ☐ 5-minute
- ☐ 60-minute

X-axis

Select a column ▼

CHART PROPERTIES

Data type

- ☐ Time series
- ☒ Summary statistics

Chart type

- ☐ Bar
- ☐ Histogram
- ☐ Line
- ☒ Scatter plot

X-axis

min_child_weight ▼

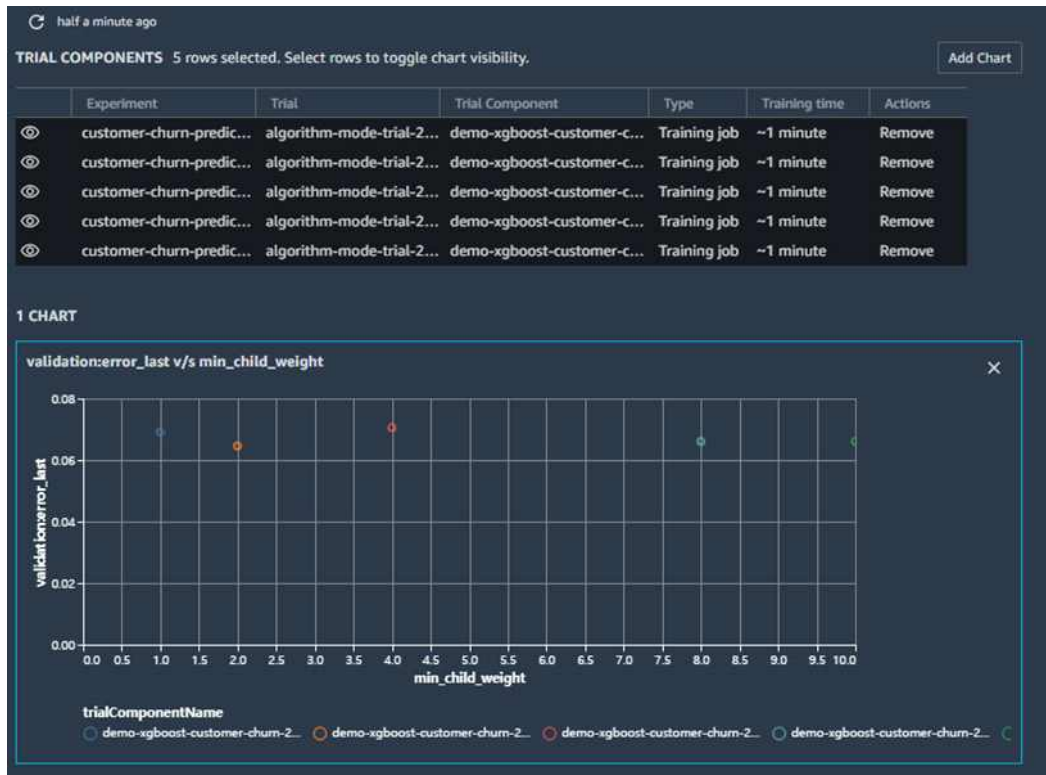
Y-axis

validation:error_last ▼

Color

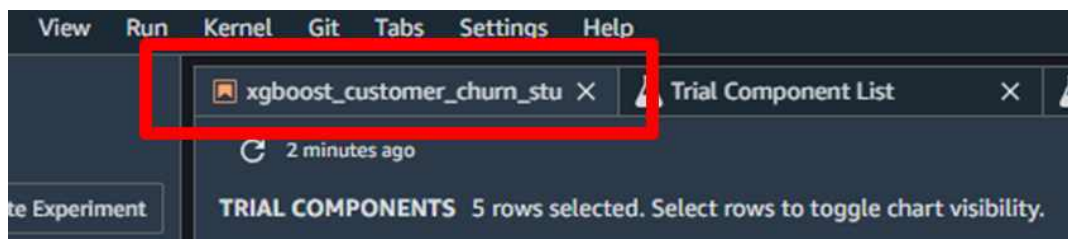
trialComponentName ▼

- 차트 모니터



2. 훈련 작업 디버깅

- 디버깅 섹션 코드 실행



- 훈련 작업을 분석하는데 사용할 규칙 지정

```
debug_rules = [Rule.sagemaker(rule_configs.loss_not_decreasing()),
                Rule.sagemaker(rule_configs.overtraining()),
                Rule.sagemaker(rule_configs.overfit())
                ]
```

```
!pygmentize xgboost_customer_churn.py
```

```

entry_point_script = "xgboost_customer_churn.py"

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

framework_xgb = sagemaker.xgboost.XGBoost(image_name=docker_image_name,
                                           entry_point=entry_point_script,
                                           role=role,
                                           framework_version="0.90-2",
                                           py_version="py3",
                                           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=sess,
                                           rules=debug_rules
                                           )

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

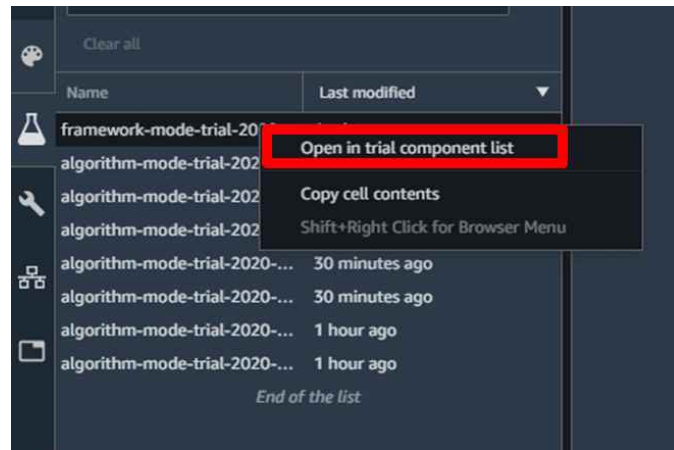
```

- 디버그 규칙을 사용하여 새 시도 생성
 - ✓ 실험 목록에서 실험 이름 두 번 클릭

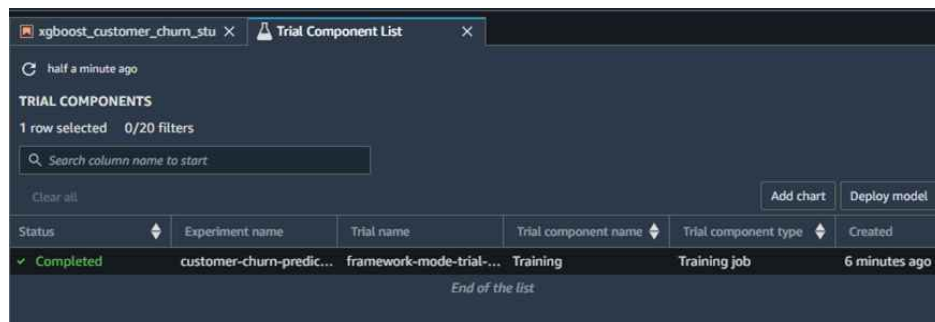
Clear all	
Name	Last modified
Unassigned trial components	
customer-churn-prediction-x...	22 minutes ago
customer-churn-prediction-x...	2 days ago
bdutest-2020-07-14-06-29-48	18 days ago
bdutest-2020-07-14-06-26-00	18 days ago
bdutest-2020-07-14-05-30-19	18 days ago
bdutest-2020-07-14-05-29-09	18 days ago

Name	Last modified
framework-mode-trial-2020-...	13 seconds ago
algorithm-mode-trial-2020-...	28 minutes ago
algorithm-mode-trial-2020-...	28 minutes ago
algorithm-mode-trial-2020-...	28 minutes ago
algorithm-mode-trial-2020-...	29 minutes ago
algorithm-mode-trial-2020-...	29 minutes ago

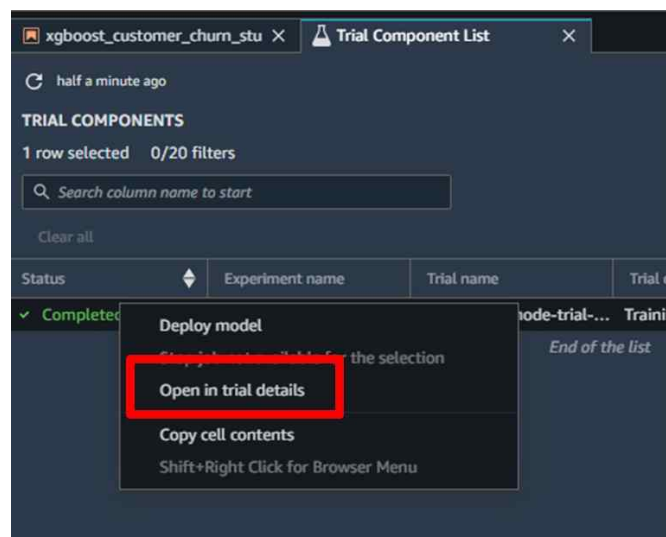
- ✓ 시도 목록에서 디버그 시도를 마우스 오른쪽 클릭하여 시도 구성 요소 목록 열기



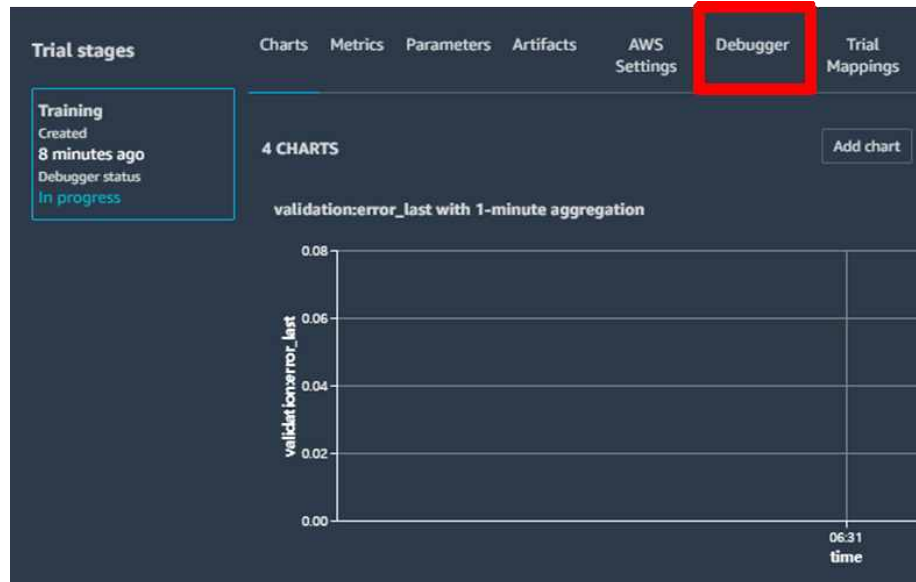
- ✓ 시도 구성 요소 목록 화면



- ✓ 시도 구성 요소 목록에서 시도를 마우스 오른쪽 버튼 클릭 후 시도세부정보 열기 클릭



- ✓ 상단 탭에서 Debugger 탭 클릭



- ✓ 디버그를 지정한 각 디버그 규칙에 대한 결과를 확인 가능

less than 5 seconds ago

Experiment: customer-churn-prediction-xgboost-2020-08-01-08-02-22

Trial: framework-mode-trial-2020-08-01-09-27-41

Status: Completed

Training Job detailed status: Completed

Stop training job

Status	Last modified	Rule name	Job ARN
No Issues Found	1 minute ago	LossNotDecreasing	arn:aws:sagemaker-east-1:4023450107...
No Issues Found	1 minute ago	Overtraining	arn:aws:sagemaker-east-1:4023450107...
No Issues Found	1 minute ago	Overfit	arn:aws:sagemaker-east-1:4023450107...

3. 모델 배포

- ✓ 배포된 모델을 호출

```
data_capture_prefix = '{}/{}/datacapture'.format(prefix)

endpoint_name = "demo-xgboost-customer-churn-" + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
print('EndpointName = {}'.format(endpoint_name))

EndpointName = demo-xgboost-customer-churn-2020-08-01-09-59-29

xgb_predictor = xgb.deploy(initial_instance_count=1,
                           instance_type='ml.m4.xlarge',
                           endpoint_name=endpoint_name,
                           data_capture_config=DataCaptureConfig(enable_capture=True,
                                                                    sampling_percentage=100,
                                                                    destination_s3_uri='s3://{}/{}'.format(bucket, data_capture_prefix)
                                                                    ))
```


- ✓ 모델을 호스팅 된 엔드포인트에 배포

```
xgb_predictor.content_type = 'text/csv'
xgb_predictor.serializer = csv_serializer
xgb_predictor.deserializer = None
```

- ✓ XGBoost 엔드 포인트를 호출하여 테스트 데이터 세트를 반복하고 예측을 수집

```
print("Sending test traffic to the endpoint {}, \nPlease wait for a minute...".format(endpoint_name))
with open('data/test_sample.csv', 'r') as f:
    for row in f:
        payload = row.rstrip('\n')
        response = xgb_predictor.predict(data=payload)
        time.sleep(0.5)
```

- ✓ 데이터 캡처

- Amazon S3에서 데이터가 캡처되었는지 확인

```
from time import sleep

current_endpoint_capture_prefix = '{}/{}'.format(data_capture_prefix, endpoint_name)
for _ in range(12): # wait up to a minute to see captures in S3
    capture_files = S3Downloader.list("s3://{}".format(bucket, current_endpoint_capture_prefix))
    if capture_files:
        break
    sleep(5)

print("Found Data Capture Files:")
print(capture_files)
```

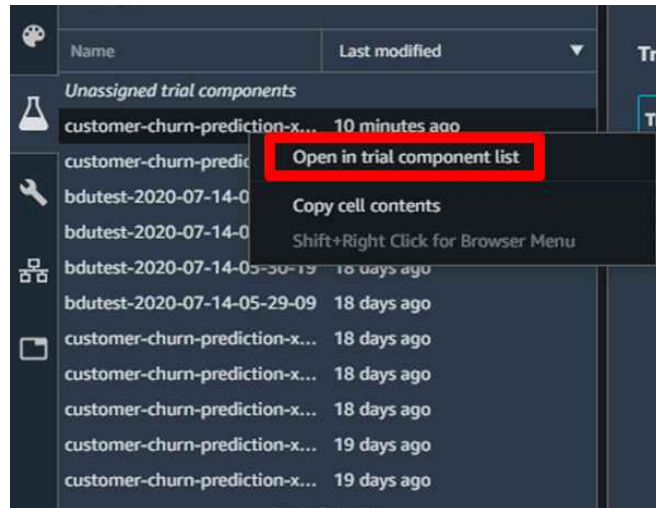
- json으로 한 줄의 내용을 빠르게 살펴 봄

```
capture_file = S3Downloader.read_file(capture_files[-1])

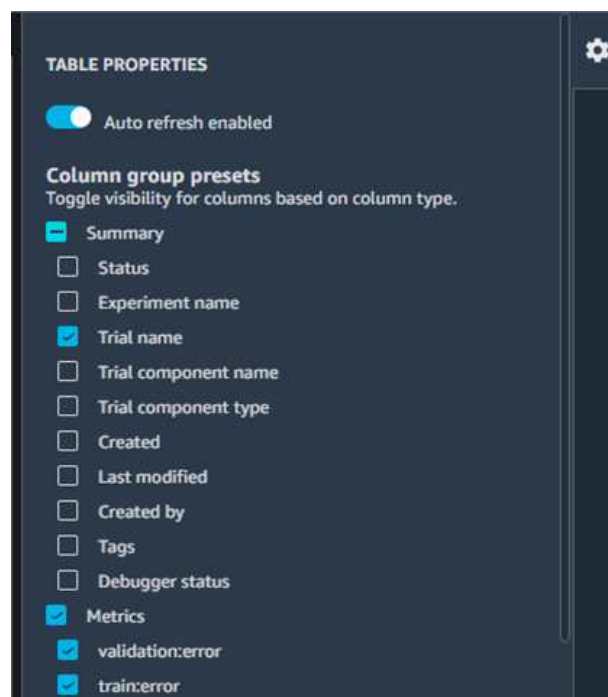
print("====Single Data Capture====")
print(json.dumps(json.loads(capture_file.split('\n')[0]), indent=2)[:2000])
```

4. 최상위 모델 배포

- ✓ 시도 구성요소 목록 진입
 - 실험 목록에서 시도명을 오른쪽 버튼으로 클릭하여 시도 구성 요소 목록 열기 선택



- ✓ 배포할 모델 선택
 - 오른쪽 상단의 환경 설정 아이콘을 선택 테이블 속성 열기
 - Summary 모두 해체 후 Trial name 체크
 - Metrics 모두 체크



✓ validation:error 헤더를 선택하여 시도 정렬

Clear all

Trial name	validation:error	train:error
framework-mode-trial-2020-08-01-09-27-41	0.072072	0.03129
algorithm-mode-trial-2020-08-01-09-00-00-weight-8	0.06606599688529968	0.02100300043821335
algorithm-mode-trial-2020-08-01-09-00-00-weight-10	0.066066	0.029147
algorithm-mode-trial-2020-08-01-08-59-59-weight-4	0.070571	0.046721
algorithm-mode-trial-2020-08-01-08-59-54-weight-1	0.069069	0.04072
algorithm-mode-trial-2020-08-01-08-59-54-weight-2	0.064565	0.025289
algorithm-mode-trial-2020-08-01-08-26-45	0.067568	0.033005

End of

Clear all

Trial name	validation:error	train:error
framework-mode-trial-2020-08-01-09-27-41	0.072072	0.03129
algorithm-mode-trial-2020-08-01-08-59-59-weight-4	0.070571	0.046721
algorithm-mode-trial-2020-08-01-08-59-54-weight-1	0.069069	0.04072
algorithm-mode-trial-2020-08-01-08-26-45	0.067568	0.033005
algorithm-mode-trial-2020-08-01-09-00-00-weight-10	0.066066	0.029147
algorithm-mode-trial-2020-08-01-09-00-00-weight-8	0.06606599688529968	0.02100300043821335
algorithm-mode-trial-2020-08-01-08-59-54-weight-2	0.064565	0.025289

End of

✓ validation:error 가 가장 낮은 시도를 마우스 오른쪽으로 클릭하고[Deploy model] 클릭

algorithm-mode-trial-2020-08-01-08-59-54-weight-1	0.069069	0.04072
algorithm-mode-trial-2020-08-01-08-26-45	0.067568	0.033005
algorithm-mode-trial-2020-08-01-09-00-00-weight-10	0.066066	0.029147
algorithm-mode-trial-2020-08-01-09-00-00-weight-8	0.06606599688529968	0.02100300043821335
algorithm-mode-trial-2020-08-01-08-59-54-weight-2	0.064565	0.025289

End of

Deploy model

Stop job not available for the selection

Open in trial details

Copy cell contents

Shift+Right Click for Browser Menu

- ✓ 엔드포인트 이름 지정
 - 호스팅할 엔드포인트 이름 지정

Deploy model

REQUIRED SETTINGS

Endpoint name

 Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.

Instance type Instance count

Data capture
 SageMaker Studio will save prediction requests and responses from the endpoint to an Amazon S3 location specified below

☐ Save prediction requests
☐ Save prediction responses

ADVANCED SETTINGS - Optional ▼

- ✓ 모델 배포
 - 설정 후 [Deploy model] 클릭

Deploy model

REQUIRED SETTINGS

Endpoint name


 Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.

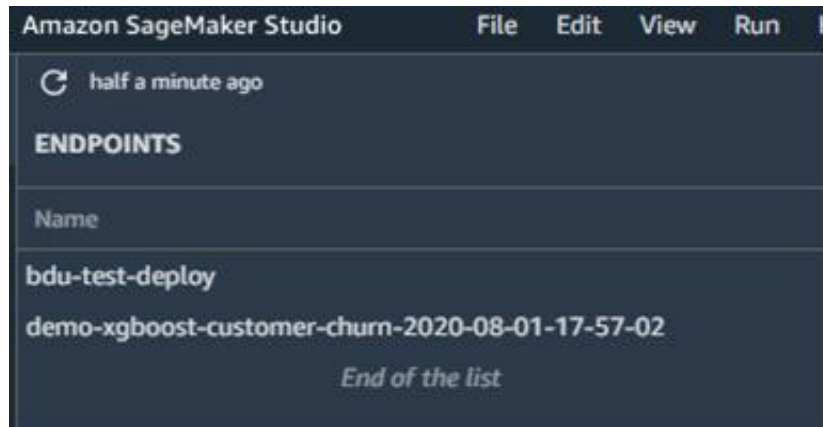
Instance type Instance count

Data capture
 SageMaker Studio will save prediction requests and responses from the endpoint to an Amazon S3 location specified below

☐ Save prediction requests
☐ Save prediction responses

Deploy model

- 스튜디오 왼쪽 탭에서 엔드포인트 리스트 아이콘() 클릭



- ✓ 스튜디오 콘솔 대시보드에서 엔드포인트 탭 클릭

엔드포인트					
<div> 엔드포인트 업데이트 작업 ▼ 엔드포인트 생성 </div> <div> <input type="text" value="엔드포인트 검색"/> </div>					
이름	ARN	생성 시간	상태	최종 업데이트 날짜	
bdu-test-deploy	arn:aws:sagemaker:us-east-1:402345010737:endpoint/bdu-test-deploy	Aug 01, 2020 18:32 UTC	Creating	Aug 01, 2020 18:32 UTC	
demo-xgboost-customer-churn-2020-08-01-17-57-02	arn:aws:sagemaker:us-east-1:402345010737:endpoint/demo-xgboost-customer-churn-2020-08-01-17-57-02	Aug 01, 2020 17:57 UTC	InService	Aug 01, 2020 18:05 UTC	

- ✓ 스튜디오 콘솔 대시보드에서 엔드포인트 구성 탭 클릭

엔드포인트 구성			
<div> 엔드포인트에 적용 복제 작업 ▼ 엔드포인트 구성 생성 </div> <div> <input type="text" value="엔드포인트 구성 검색"/> </div>			
이름	ARN	생성 시간	
bdu-test-deploy-endpointConfig-1596306732110	arn:aws:sagemaker:us-east-1:402345010737:endpoint-config/bdu-test-deploy-endpointconfig-1596306732110	Aug 01, 2020 18:32 UTC	
demo-xgboost-customer-churn-2020-08-01-17-57-02	arn:aws:sagemaker:us-east-1:402345010737:endpoint-config/demo-xgboost-customer-churn-2020-08-01-17-57-02	Aug 01, 2020 17:57 UTC	

5. 리소스 정리

- ✓ 소스 코드로 리소스 정리

```

try:
    sess.delete_monitoring_schedule(mon_schedule_name)
except:
    pass
while True:
    try:
        print("Waiting for schedule to be deleted")
        sess.describe_monitoring_schedule(mon_schedule_name)
        sleep(15)
    except:
        print("Schedule deleted")
        break

sess.delete_endpoint(xgb_predictor.endpoint)

def cleanup(experiment):
    """Clean up everything in the given experiment object"""
    for trial_summary in experiment.list_trials():
        trial = Trial.load(trial_name=trial_summary.trial_name)

        for trial_comp_summary in trial.list_trial_components():
            trial_step = TrialComponent.load(trial_component_name=trial_comp_summary.trial_component_name)
            print("Starting to delete trial component... " + trial_step.trial_component_name)
            sm.disassociate_trial_component(TrialComponentName=trial_step.trial_component_name, TrialName=trial_name)
            trial_step.delete()
            time.sleep(1)

        trial.delete()

    experiment.delete()

cleanup(customer_churn_experiment)

```

- ✓ 콘솔에서 리소스 정리

- aws 콘솔에서 모델, 엔드포인트 구성, 엔드포인트 등 재 확인

▼ 추론

컴파일 작업

모델 패키지

모델

엔드포인트 구성

엔드포인트

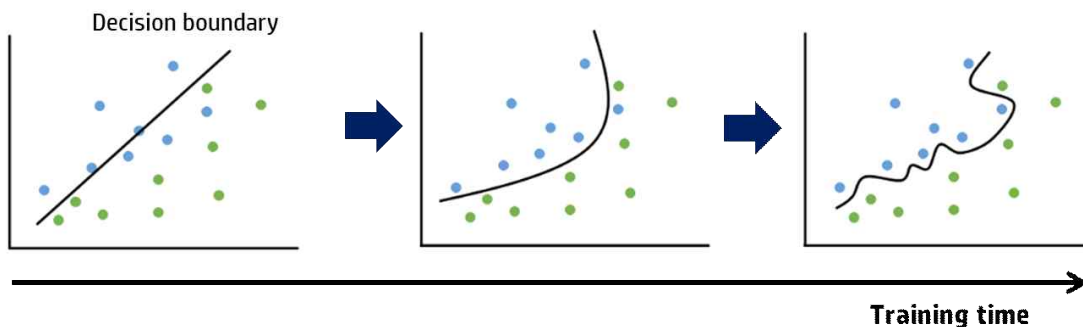
배치 변환 작업

- ✓ S3 콘솔 접속 후 버킷 삭제

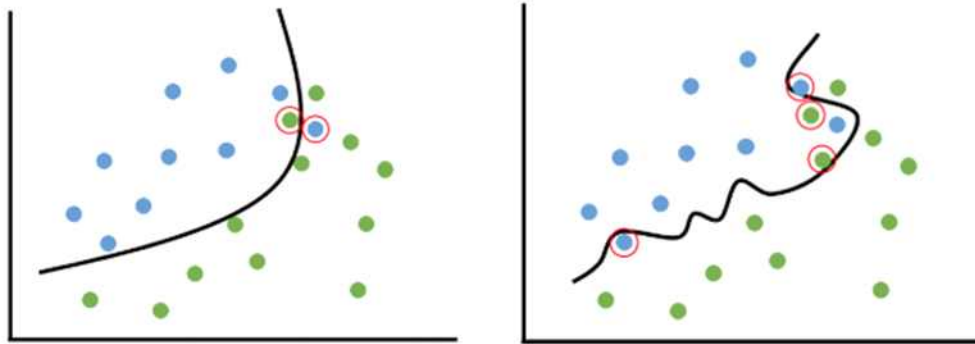


※ 나도 전문가다

- 과소적합(underfitting)과 과적합(overfitting)
 - ✓ 머신러닝의 궁극적인 목표는 훈련 데이터 세트를 이용하여 학습한 모델을 가지고 테스트 데이터 세트를 예측하는 것
 - ✓ 이때 테스트 데이터 세트는 학습 과정에서 참조할 수 없다고 가정하기 때문에 머신 러닝 모델은 훈련 데이터세트만 가지고 테스트 데이터세트를 잘 예측하도록 학습되어야 함



- ✓ 머신 러닝 모델은 training loss가 작아지도록 학습을 진행하기 때문에 학습이 진행 될수록 모델의 decision boundary는 점점 training dataset에 fitting 됨



- ✓ 훈련 데이터세트와 테스트 데이터 세트가 정확히 일치할 경우, 훈련 데이터 세트에 fitting될 수록 모델의 예측도는 증가
- ✓ 하지만 실제 훈련 데이터세트와 테스트 데이터세트는 다른 경우가 많으며, 그림과 같이 훈련 데이터세트에 조금은 덜 fitting된 모델이 테스트 데이터 세트에 더 높은 정확도를 보일 수 있음



- ✓ 우리의 목적은 학습을 통해 머신 러닝 모델의 underfitting된 부분을 제거해 나가면서 overfitting이 발생하기 직전에 학습을 멈추는 것

평가하기

1. 세이지메이커 스튜디오에서 엔드포인트 정보를 확인하기 위하여 선택해야 할 아이콘을 고르시오.



- 정답 : ①번

해설 : 세이지메이커 엔드포인트 목록 아이콘은 배포된 모델의 엔드포인트 목록을 제공합니다.

2. 훈련 작업을 분석하는데 사용할 규칙을 지정하여 시간, 리소스 및 비용을 줄이기 위한 작업은?

- ① 데이터 시각화
- ② 모델 배포
- ③ 훈련 작업 디버깅
- ④ 리소스 정리

- 정답 : ③번

해설 : 디버거를 사용하면 모델 훈련에 필요한 시간, 리소스 및 비용을 획기적으로 줄일 수 있습니다.

학습정리

1. 훈련 결과 시각화

- 시각화할 시도(Trial) 선택
- 차트 속성 구성
- 차트 모니터

2. 훈련 작업 디버깅

- 훈련 작업 분석 규칙 지정
- 디버그 규칙을 사용하여 새시도 생성
- 디버그 결과 확인

3. 모델 배포

- 모델 호스팅
- 배포된 모델 호출
- 데이터 캡처

4. 최상의 모델 배포

- 시도 구성 요소 목록 진입
- 배포할 모델 선택
- 엔드 포인트 선택
- 모델 배포

5. 리소스 정리

- 소스 코드로 리소스 정리
- 콘솔에서 리소스 정리