7. 세이지메이커 노트북 인스턴스

3강. 학습 모델 구축 실습

학습목표

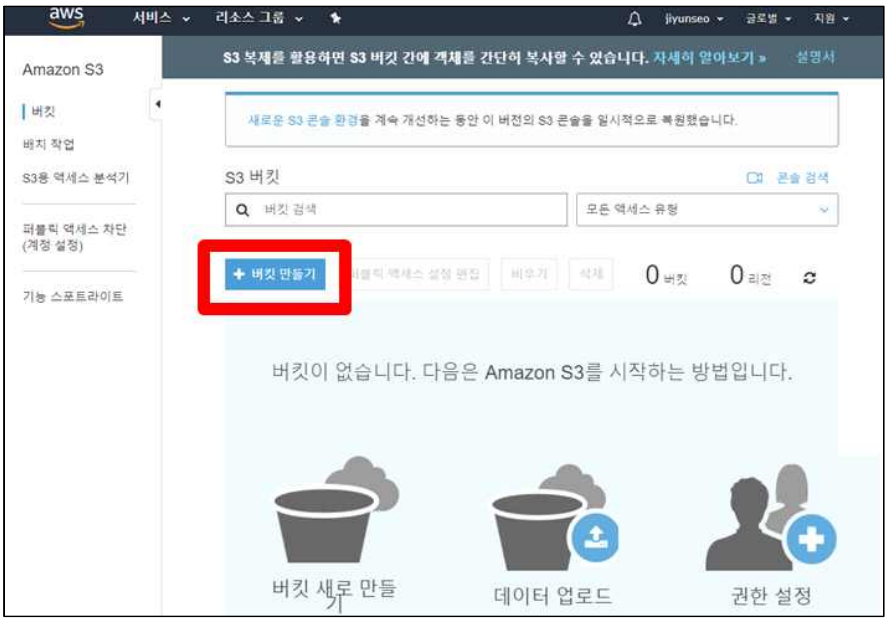
- 세이지메이커 노트북 인스턴스를 이용하여실제 머신러닝 학습 모델을 구축할 수 있다.

학습내용

- Amazon S3 버킷 생성
- 세이지메이커 노트북 인스턴스 및 주피터 노트북 생성
- 데이터 생성
- 모델 훈련
- 모델 배포 및 검증
- 정리

1. Amazon S3 버킷 생성

- S3 콘솔 접속 후 버킷 만들기 클릭



- 버킷 정보 입력 후 버킷 생성(리전은 sagemaker 서비스 사용가능한 리전으로 생성)완료 후 버킷 이름 클릭

- 폴더 만들기 클릭

- 폴더 이름을 입력 후 저장 클릭

- 폴더 생성 완료

2. 세이지메이커 노트북 인스턴스 및 주피터 노트북 생성

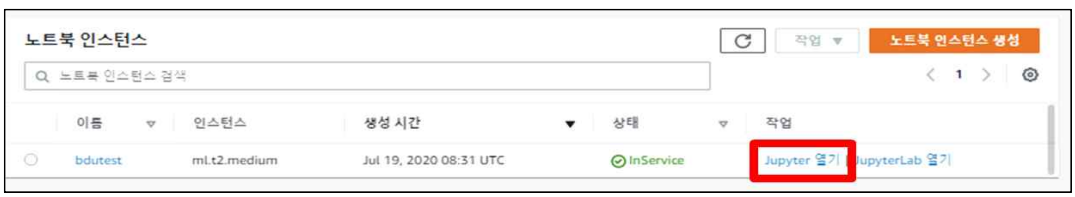
- sagemaker 콘솔 접속 후 노트북 인스턴스 클릭

- 노트북 인스턴스 생성 클릭

- 노트북 인스턴스 설정 정보 입력 후 [노트북 인스턴스 생성] 클릭

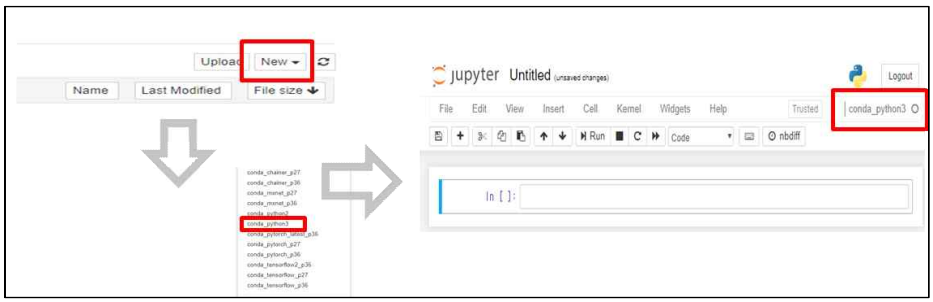
- 노트북 인스턴스 생성이 완료 되면 pending 상태로 노트북 인스턴스가 생성 됨

- [jupyter 열기] 클릭



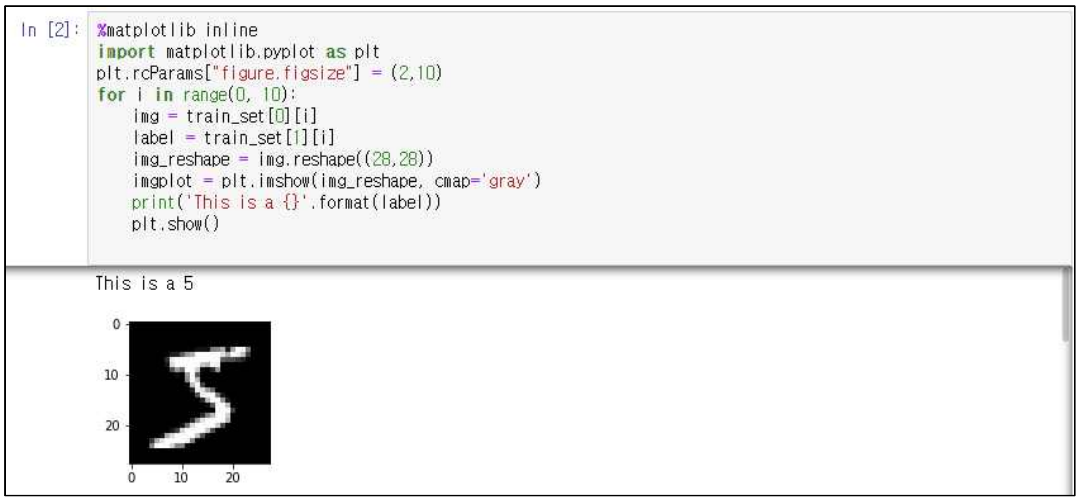
3. 데이터 생성

- 노트북 생성
  - ✓ 파일 탭에서 새로 만들기를 선택하고 conda_python3를 선택
  - ✓ 사전 설치된 환경에는 기본 Anaconda 설치, Python3이 포함되어 있음



- MNIST Database 웹 사이트에서 노트북으로 데이터세트를 다운로드
  - ✓ 데이터 세트 파일명 : mnist.pkl.gz

- Train_set에서 처음 10개의 이미지를 표시



- 라이브러리 선언

```
In [3]: %%time
        import os
        import boto3
        import re
        import copy
        import time
        import io
        import struct
        from time import gmtime, strftime
        from sagemaker import get_execution_role

        CPU times: user 687 ms, sys: 97.4 ms, total: 784 ms
        Wall time: 7.83 s
```

- 규칙, 리전, 버킷이름, 데이터를 저장하는 버킷 경로

  ✓ myBucket = 생성된 버킷 이름, sagemaekr = 버킷에 생성한 폴더 이름

```
In [4]: role = get_execution_role()

        region = boto3.Session().region_name

        bucket='bdu03'
        prefix = 'sagemaker'
```

- 데이터 세트 형식을 numpy.array 형식에서 CSV 형식으로 변환

```
def convert_data():
    data_partitions = [('train', train_set), ('validation', valid_set), ('test', test_set)]
    for data_partition_name, data_partition in data_partitions:
        print('{}: {} {}'.format(data_partition_name, data_partition[0].shape, data_partition[1].shape))
        labels = [t.tolist() for t in data_partition[1]]
        features = [t.tolist() for t in data_partition[0]]

        if data_partition_name != 'test':
            examples = np.insert(features, 0, labels, axis=1)
        else:
            examples = features
        #print(examples[50000,:])


        np.savetxt('data.csv', examples, delimiter=',')



        key = "{}/{}/examples".format(prefix,data_partition_name)
        url = 's3://{}/{}'.format(bucket, key)
        boto3.Session().resource('s3').Bucket(bucket).Object(key).upload_file('data.csv')
        print('Done writing to {}'.format(url))

convert_data()
```

```
train: (50000, 784) (50000,)
Done writing to s3://bdu03/sagemaker/train/examples
validation: (10000, 784) (10000,)
Done writing to s3://bdu03/sagemaker/validation/examples
test: (10000, 784) (10000,)
Done writing to s3://bdu03/sagemaker/test/examples
```

4. 모델 훈련

- Amazon SageMaker Python SDK 및 XGboost 컨테이너를 가져옴

```
import sagemaker

from sagemaker.amazon.amazon_estimator import get_image_uri

container = get_image_uri(boto3.Session().region_name, 'xgboost')
```

```
'get_image_uri' method will be deprecated in favor of 'ImageURIProvider' class in SageMaker Python SDK v2.
WARNING:root:There is a more up to date SageMaker XGBoost image. To use the newer image, please set 'repo_version'='1.0-1'. For example:
        get_image_uri(region, 'xgboost', '1.0-1').
```

- 데이터를 업로드한 s3 위치에서 훈련 및 검증 데이터를 다운로드하고 훈련 출력을 저장할 위치를 설정

```
In [5]: train_data = 's3://{}/{}/{}'.format(bucket, prefix, 'train')

        validation_data = 's3://{}/{}/{}'.format(bucket, prefix, 'validation')

        s3_output_location = 's3://{}/{}/{}'.format(bucket, prefix, 'xgboost_model_sdk')
        print(train_data)
```

```
s3://bdu03/sagemaker/train
```

- sagemaker.estimator.Estimator 클래스의 인스턴스를 생성

```
xgb_model = sagemaker.estimator.Estimator(container,
                                          role,
                                          train_instance_count=1,
                                          train_instance_type='ml.m4.xlarge',
                                          train_volume_size = 5,
                                          output_path=s3_output_location,
                                          sagemaker_session=sagemaker.Session())
```

```
WARNING:root:Parameter image_name will be renamed to image_uri in SageMaker Python SDK v2.
```

- set_hyperparameters 메서드를 호출하여 XGBoost 훈련 작업의 하이퍼파라미터 값을 설정

```
xgb_model.set_hyperparameters(max_depth = 5,
                              eta = .2,
                              gamma = 4,
                              min_child_weight = 6,
                              silent = 0,
                              objective = "multi:softmax",
                              num_class = 10,
                              num_round = 10)
```

- 훈련 작업에 사용할 훈련 채널을 생성

```
train_channel = sagemaker.session.s3_input(train_data, content_type='text/csv')
valid_channel = sagemaker.session.s3_input(validation_data, content_type='text/csv')

data_channels = {'train': train_channel, 'validation': valid_channel}
```

```
WARNING:sagemaker:'s3_input' class will be renamed to 'TrainingInput' in SageMaker Python SDK v2.
WARNING:sagemaker:'s3_input' class will be renamed to 'TrainingInput' in SageMaker Python SDK v2.
```

- 모델 훈련을 시작하려면 예측기의 fit 메소드를 호출

```
xgb_model.fit(inputs=data_channels,  logs=True)
```

```
h=5
[09:02:07] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 50 extra nodes, 8 pruned nodes, max_depth
=5
[09:02:07] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 48 extra nodes, 8 pruned nodes, max_depth
=5
[09:02:08] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 52 extra nodes, 6 pruned nodes, max_depth
=5
[09:02:08] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 38 extra nodes, 14 pruned nodes, max_dept
h=5
[09:02:09] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 60 extra nodes, 2 pruned nodes, max_depth
=5
[09:02:10] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 50 extra nodes, 6 pruned nodes, max_depth
=5
[9]#011train-merror:0.06942#011validation-merror:0.0773

2020-07-19 09:02:19 Uploading - Uploading generated training model
2020-07-19 09:02:19 Completed - Training job completed
Training seconds: 146
Billable seconds: 146
```

5. 모델 배포 및 검증

- deploy 메서드가 배포 가능한 모델을 생성하고, Amazon SageMaker 호스팅 서비스 엔드포인트를 구성하고, 모델을 호스팅할 엔드포인트를 시작

```
In [11]: xgb_predictor = xgb_model.deploy(initial_instance_count=1,
                                          content_type='text/csv',
                                          instance_type='ml.t2.medium'
                                          )
```

```
WARNING:sagemaker:Parameter image will be renamed to image_uri in SageMaker Python SDK v2.
---------------!
```

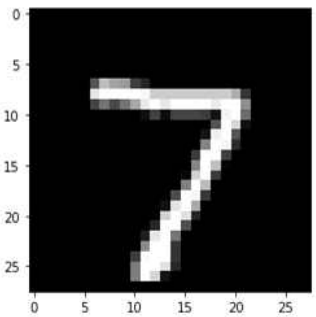
- Amazon S3에서 테스트 데이터를 다운로드

```
s3 = boto3.resource('s3')

test_key = "{}/test/examples".format(prefix)

s3.Bucket(bucket).download_file(test_key, 'test_data')
```

- 테스트 데이터 세트의 처음 10개 이미지를 레이블로 플롯

```
%matplotlib inline

for i in range (0, 10):
    img = test_set[0][i]
    label = test_set[1][i]
    img_reshape = img.reshape((28,28))
    imgplot = plt.imshow(img_reshape, cmap='gray')
    print('This is a {}'.format(label))
    plt.show()
```

```
This is a 7
```



```
This is a 2
```

- 테스트 데이터 세트의 처음 10개 예제에 대해 추론을 가져오기

```
with open('test_data', 'r') as f:
    for j in range(0,10):
        single_test = f.readline()
        result = xgb_predictor.predict(single_test)
        print(result)
```

```
b'7.0'
b'2.0'
b'1.0'
b'0.0'
b'4.0'
b'1.0'
b'4.0'
b'9.0'
b'5.0'
b'9.0'
```

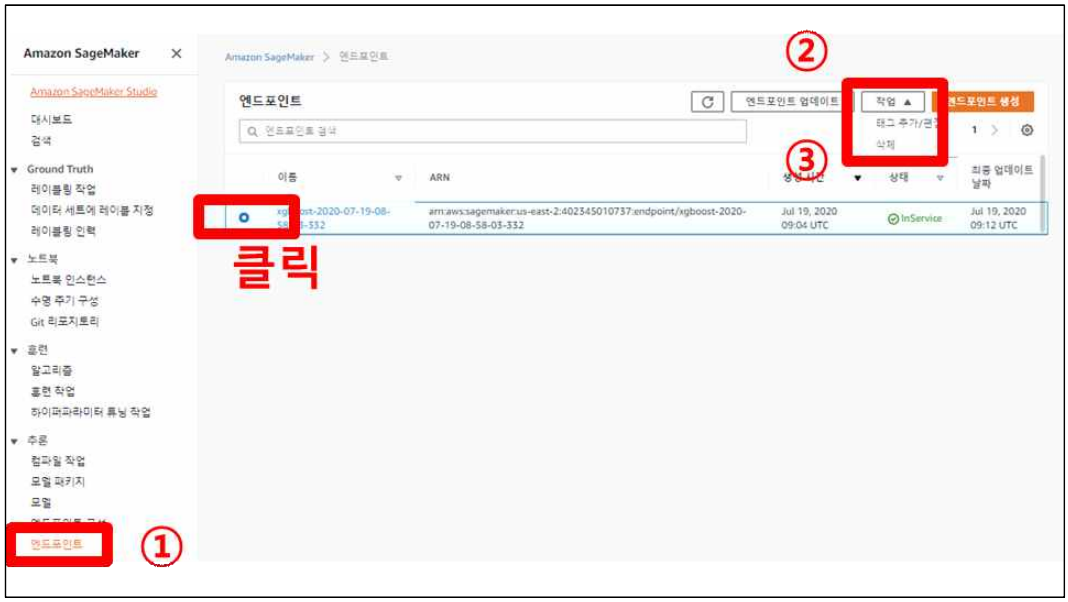
5. 정리

- 엔드포인트 삭제

  ① 추론에서 엔드포인트를 선택

  ② 예제에서 생성한 엔드폰인트를 선택한 다음 action(작업)Delete(삭제 선택)

  ③ Delete를 선택



- 엔드포인트 구성 삭제

  ① 추론에서 엔드포인트 구성을 선택

  ② 예제에서 생성한 앤드포인트 구성을 선택한 다음, action(작업)/Delete(삭제 선택)

  ③ Delete를 선택

- 모델 삭제

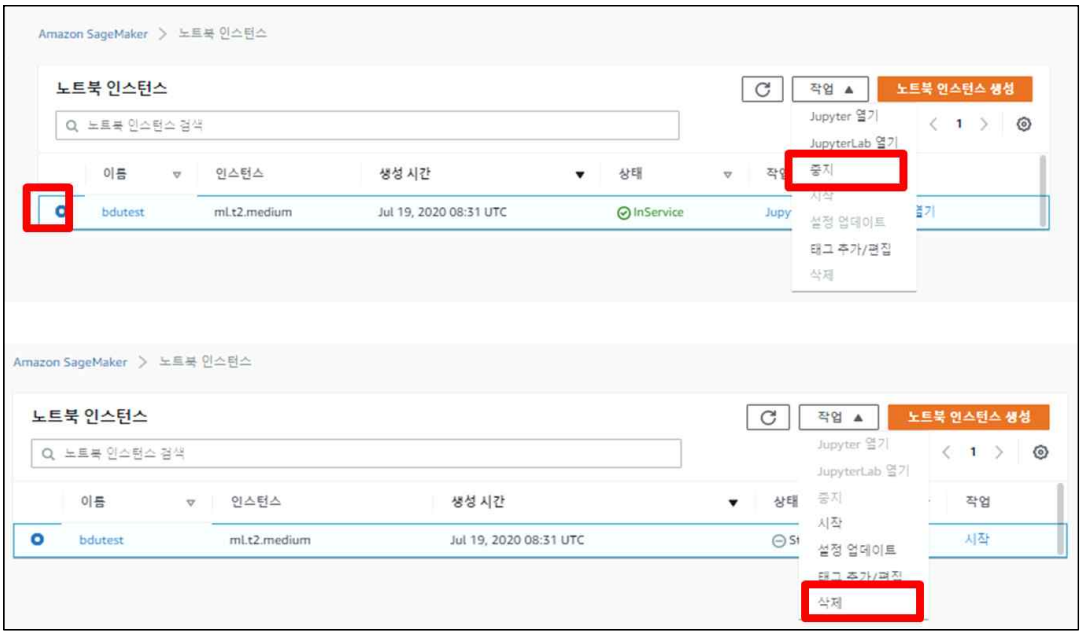
  ① 추론에서 모델을 선택

  ② 예제에서 생성한 모델을 선택한 다음, action(작업)Delete(삭제 선택)

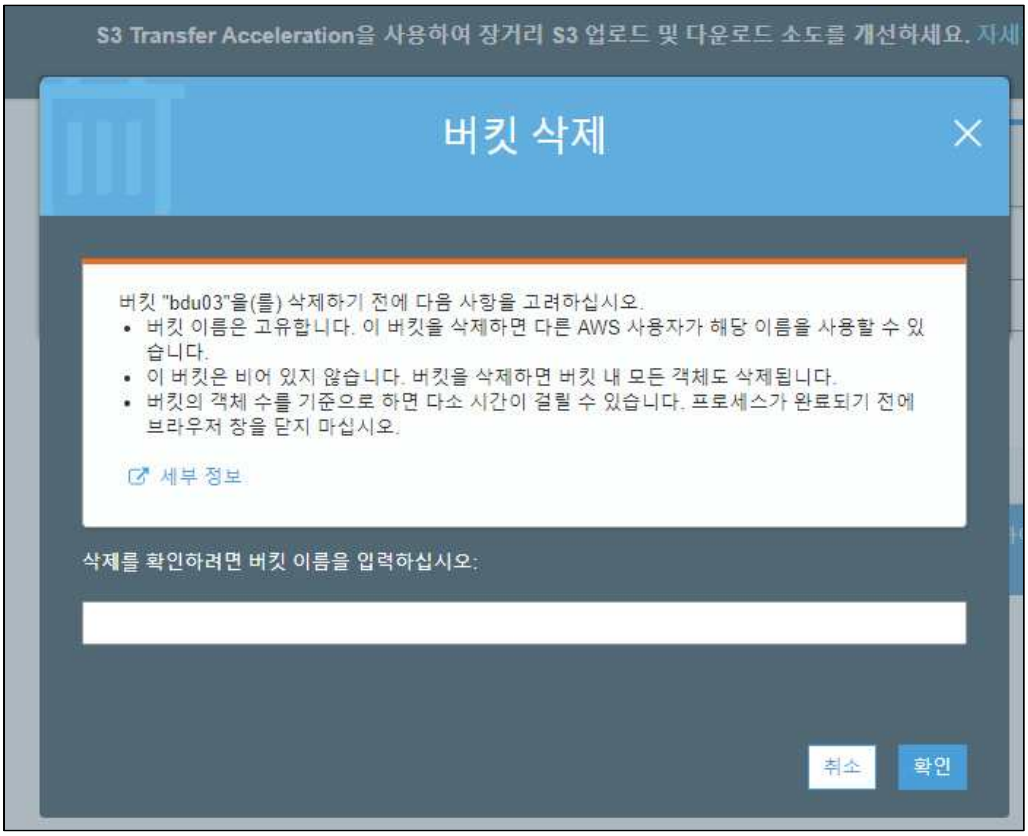
  ③ Delete를 선택

- 노트북 인스턴스 삭제

  ✓ 삭제할 노트북 인트턴스 클릭 후 작업 탭에서 중지 후 삭제

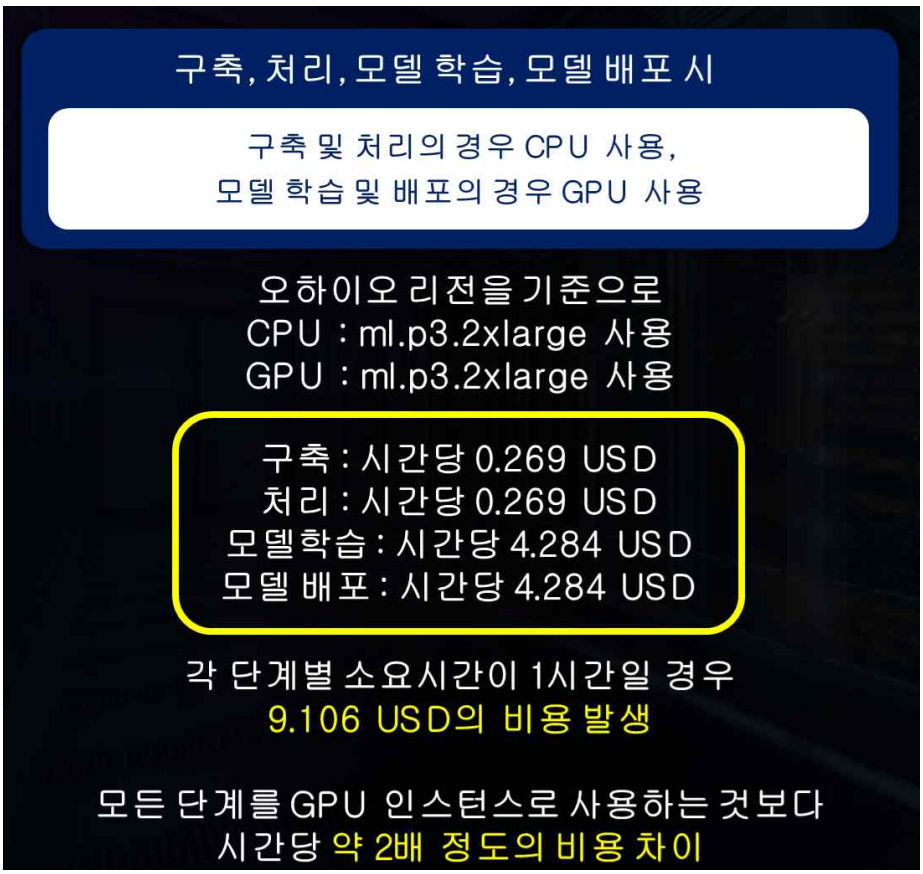


- S3 삭제
  - ✓ 삭제할 버킷 클릭 삭제 클릭
  - ✓ 버킷 내부에 파일이 있을 경우 비우기 작업을 먼저 후 삭제



※ 나도 전문가다

- SageMaker 인스턴스별 요금 비교



평가하기

1. **실습에서 데이터를 생성하는 단계와 관련 없는 것은?**

   ① 주피터 노트북 생성

   ② MNIST 데이터세트 다운로드

   ③ 훈련 데이터 세트 탐색

   ④ 훈련 데이터 세트 변환 및 S3에 업로드

   - **정답 : ①번**

   해설 : 주피터 노트북 생성은 데이터 생성단계로보기 어렵습니다.

2. **빈칸에 알맞은 단어를 고르시오.**

   보기 : fit, CSV, 테스트

   ① 데이터 생성단계에서 데이터 세트 형식을numpy.array 형식에서 [ ] 형식으로 변환한다.

   ② 모델 배포 후, 검증을 위해 Amazon S3에서 [ ] 데이터를 다운로드 한다.

   ③ 모델 훈련을 시작하기 위해서 [ ] 메소드를 호출한다.

   - **정답 : ① CSV, ② 테스트, ③ fit**

학습정리

1. **Amazon S3 버킷 생성**
   - 데이터세트, 모델 등을 저장하기 위하여 사용
2. **세이지메이커 노트북 인스턴스 및 주피터 노트북 생성**
   - 모델을 학습하기 위한 컴퓨팅 환경과 개발환경 구축
3. **데이터 생성**
   - 학습 모델을 구축하기 위한 데이터셋트 생성 과정
4. **모델 훈련**
   - 학습 모델 훈련 과정
5. **모델 배포 및 검증**
   - 애플리케이션에서 모델을 사용하기 위한 배포 및 검증 단계
6. **정리**
   - 불필요한 비용을 방지하기 위하여 사용한 리소스를 삭제